UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

~ Technical University of Catalonia & Higher Institute of Aeronautics and Space ~

Double Degree Master's Thesis

Design and Automation of Cybersecurity and Performance Validation Test Cases Applied to Connected Vehicles

---

In partial fulfilment of the requirements for the degrees of

Màster en Enginyeria de Telecomunicació (MSc) & Ingénieur ISAE-SUPAERO (MSc)

Carried out at Renault Software Labs (RSWL), a fully-owned subsidiary of

GROUPE RENAULT

Author: Luis ORUS GRACIA

Advisor at RSWL: Dominique GRANGE

Advisor at UPC-ETSETB: José Antonio LAZARO VILLA

Advisor at ISAE-SUPAERO: Jérôme LACAN

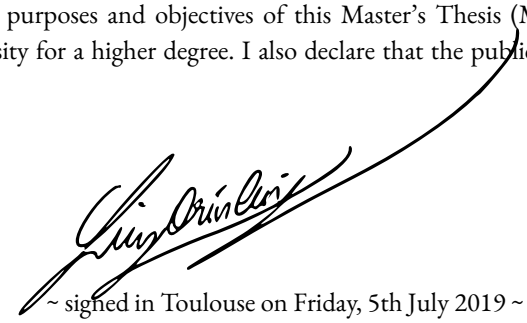
Barcelona ~ July 2019

Title of the Master's Thesis
Design and Automation of Cybersecurity and Performance Validation Test Cases Applied to Connected Vehicles

Titre du Projet de Fin d'Études
Définition d'un environnement de test de cybersécurité et de robustesse appliqué aux véhicules connectés

Título del Trabajo de Fin de Máster
Diseño y automatización de tests de validación de ciberseguridad y rendimiento aplicados a los vehículos conectados

Statement of Authenticity
I, the undersigned and author, Luis ORUS GRACIA, declare that this dissertation is my original work, gathered and utilized especially to fulfil the purposes and objectives of this Master's Thesis (MSc), and has not been previously submitted to any other university for a higher degree. I also declare that the publications cited in this work have been personally consulted.



~ signed in Toulouse on Friday, 5th July 2019 ~

Advisors
Dominique GRANGE, José Antonio LAZARO VILLA, Jérôme LACAN

Evaluation Board

President	Lluis JOFRE ROCA, PhD Department of Signal Theory and Communications (TSC) jofre@tsc.upc.edu
Member	Juan Luis GORRICHO MORENO, PhD Department of Telematics Engineering (ENTEL) juanluis@entel.upc.edu
Secretary	José Antonio LAZARO VILLA, PhD Department of Signal Theory and Communications (TSC) jose.lazaro@tsc.upc.edu
Date	Tuesday, 16th July 2019
Hour	10:30 am
Location	Sala Multimèdia, Building B3, ETSETB, UPC Campus Nord, Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

A Día, mi modelo a seguir.

"La inspiración existe, pero tiene que encontrarte trabajando"
~ Pablo PICASSO

Acknowledgements

First and foremost I would like to thank Dominique, my project advisor at Renault Software Labs (RSWL). He was patient with me, and gave me reasonable freedom when deciding what to do next at every step of the project. He also showed me that life rewards honest, hard-working, good people. For that, I am thankful. He taught me that there is no little ambition, but little work. The more you invest in life, the more rewards you get. He also helped me improving my French, for I know that even now, after two years living in France, it is definitely not perfect. He taught me French history and culture, allowing me to fully immerse within the country. He even taught me about "La Nueve", a French Armored Division composed by many Spanish republicans that played an important role in the Liberation of Paris in 1944. *Merci Dom*.

I would like to thank Cyril for being such a supportive coworker. He greatly enlivened my stay in the laboratory, even though I did not know most of the people he talked about. Old people, you know. *Mais c'est qui ce Dominique Valera, hein ? Arnold ? Ah oui, lui je connais !*

I would like to thank both Amaury and Eric, my managers. They supported all my work and always encouraged me to do my best. *Messieurs Chefs, je vous en remercie*.

My special thanks to Guillaume, David, Pierre, Fernand and Dominique for taking the time to review this document.

I would like to thank all of my coworkers at RSWL: Vernon, Mickael, Remy 1 – *A Cuenca!*, Remy 2, Franck, Helia, Rihab, Jérôme, Thomas. But also thanks to the Spanish Armada: Pepe, Manu, Juan.

All of them have made me considering France as my second home. To me, it is a welcoming country that values foreigners and invest in people no matter who they are or where they come from. *Merci à tous pour ces instants de bonheur. Vous venez de collaborer dans la création d'un européen convaincu. Je suis espagnol, mais je me sens plus français chaque jour. Vive la France qui accueille les étrangers ! Ne perdez pas vos valeurs républicaines. Je me battrai pour elles pendant toute ma vie, en Espagne ou d'ailleurs*.

My many thanks to RSWL, UPC-ETSETB, and ISAE-SUPAERO.

My distinguished thanks to my family and close friends. It's a little cliché, but thank you anyways. Thank you all for your immeasurable support.

Last, but not least, my special thanks to my sister, Día. **She shines her light on the road ahead**.

Abstract

We are living today the so called “Digital Revolution” that started 50 years ago. Younger generations are forcing the transportation industry to move towards social, connected, environment-friendly, assisted means of transport. The intelligent, fully-digitized autonomous vehicle is an evolutionary process that starts right into the connected vehicle. But such complex system is becoming more and more vulnerable to external cyber attacks as it increases its connectivity capabilities. However, the Groupe Renault, the world’s leading French vehicle-manufacturer, is ready to confront these challenges. This project focuses on the design, implementation and automation of a variety of test cases aiming to partially validate the embedded software running on the IVC, the on-board modem that provides Internet accessibility to connected vehicles. Both cybersecurity and performance aspects will be considered, a series of results will be presented and analyzed, and some relevant conclusions will be eventually drawn.

This project is part of the Dual Degree Program arranged between the UPC-ETSETB –based in Barcelona and specialized in Telecommunications, and ISAE-SUPAERO –based in Toulouse and specialized in Aeronautics and Space, two of the best European Engineering Schools. It was carried out at Renault Software Labs, a fully-owned subsidiary of the Groupe Renault, based in Toulouse, France.

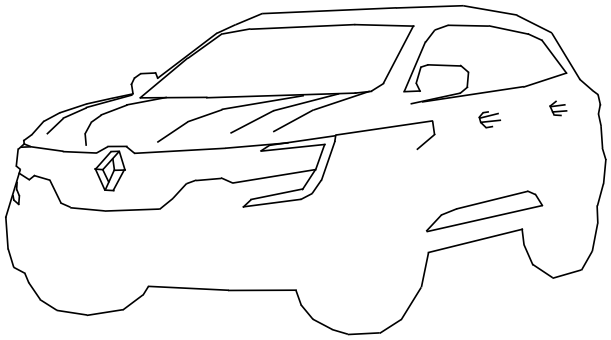


Table of contents

List of acronyms

3G Third Generation.
4G Fourth Generation.
5G Fifth Generation.

ADAS Advanced Driver-Assistance Systems.
AI Artificial Intelligence.
AOSP Android Open Source Project.

BCM Brake Control Module.
BS Base Station.

CA Certificate Authority.
CAN Controller Area Network.
CCAR Connected Car.
CI Continuous Integration.
CID Center Information Display.
CMF Common Module Family.
CSR Certificate Signing Request.
CTF Capture The Flag.

DAU Data Application Unit.
DCM Data Connection Manager.
DDoS Distributed Denial-of-Service.
DF Development Framework.
DoS Denial-of-Service.

E2E End-to-end.
EBA Emergency Brake Assist.
ECU Engine Control Unit.
ETSETB Escola Tècnica Superior d'Enginyeria de Telecomunicacions de Barcelona, or Barcelona School of Telecommunications Engineering, in English.

FCA Fiat Chrysler Automobiles.
FOTA Firmware Over-The-Air.

GAS Google Automotive Services.
GDPR General Data Protection Regulation.
GPIB General Purpose Interface Bus.
GPS Global Positioning System.

GSM Global System for Mobile Communications.

HAL Hardware Abstraction Layer.
HUD Head-Up Display.

ICCID Integrated Circuit Card IDentifier.
ID Identity Document.
IMSI International Mobile Subscriber Identity.
IoT Internet of Things.
IP Intellectual Property.
IP Internet Protocol.
ISAE-SUPAERO Institut Supérieur de l'Aéronautique et de l'Espace, or Higher Institute of Aeronautics and Space, in English.
ITU International Telecommunication Union.
IVC In-Vehicle Communication.
IVI In-Vehicle Infotainment.

LTE Long-Term Evolution.

MaaS Mobility-as-a-Service.
MATRIX Micro-services Automotive Test Robot Integration eXecutor.
MCG Mobile Communication Group.
MITM Man-In-The-Middle.
ML Machine Learning.
ML Reinforcement Learning.

OEM Original Equipment Manufacturer.
OS Operating System.

PCA Principal Component Analysis.
PIN Personal Identification Number.
PKI Public Key Infrastructure.
PnP Power and Performance.
PO Product Owner.
PS Packet Switched.
PUK Personal Unblocking Key.

QA Quality Assurance.

R&D Research and development.
RF Radio Frequency.
RHL Remote Horn and Lights.
RKIS Remote Keyless Ignition System.
RKS Remote Keyless System.
RLU Remote Lock Unlock.
RNM Renault-Nissan-Mitsubishi.
RRC Radio Resource Control.
RSA Rivest-Shamir-Adleman.

RSM Renault Samsung Motors.
RSWL Renault Software Labs.

SBL Software Business Line.
SCM Suspension Control Module.
SHA Secure Hash Algorithm.
SIM Subscriber Identification Module.
SIT Software Integration and Test.
SMTM Software Mainline and Test Management.
SOTA Software Over-The-Air.

TC Test Case.
TCP Transmission Control Protocol.
TCR Technocentre Renault.
TLS Transport Layer Security.
TTC Time-To-Connection.

UDP User Datagram Protocol.
UMTS Universal Mobile Telecommunications System.
UPC Universitat Politècnica de Catalunya, or Technical University of Catalonia, in English.
UX User Experience.

VCS Version Control Systems.
VIN Vehicle Identification Number.

WCDMA Wideband Code Division Multiple Access.
WPAN Wireless Personal Area Networks.

Part I

Project context

Chapter 1

Introduction

1.1 The new digital revolution: towards connected vehicles

We are living today the so called "Digital Revolution" that started over 50 years ago. A natural evolution process that once started with the shift from mechanical and analogue electronic devices towards digital systems. We first witnessed the expansion of the dot-com websites in the 1990s and its subsequent bubble in the 2000s. We also gave testimony to the later triumph and development of mobile devices with the pioneer arrivals of the BlackBerry, released in 1999, and the widescreen iPhone, released in 2007. It then started the unstoppable smartphone industry revolution towards the socialization of digital telephony that we can find at present-day. The "digital world" as a whole is changing the way we interact with each other as a society, and forcing every business to rapidly innovate and adapt to the fast changing work market where both outdated professionals and enterprises are left apart to die alone or to be eaten by huge digital multinationals. It is now when Google, Amazon and Facebook are being the target of an exhausted scrutiny on the part of the competent authorities that suggest the possible existence of a technological monopoly [1]. Leaving the controversy regarding honest competition aside, we observe how there is clearly a lack of digital transformation in something that affects us daily: transportation. Younger generations are forcing the transportation industry to move towards social, connected, environment-friendly means of transport. Electric and hybrid vehicles are a reality and are here to stay. Global warming is certain and without the required changes towards a digital transformation, the automobile industry will cease to exist. Websites in the late 1990s. Smartphones in the late 2010s. *Smartcars* in the late 2020s. It is the natural process of digital transformation, and the Groupe Renault[1] is trying to push it hard in order to offer a wide range of vehicles that introduce both eco-friendly engines and connected services. We say that the intelligent vehicle, or smartcar, is an evolutionary process that is composed by several phases, starting from the connected vehicle until the arrival of the autonomous vehicle.

The initial step consists of a vehicle that connects via the Internet to different servers in order to offer an ample range of services or actions, from remotely opening the door –the so called Remote Lock Unlock (RLU) to on-board computer control by voice. It also offers a full range of infotainment systems, optimized driving modes to obtain the lowest possible fuel consumption and battery-life

[1]Name in French. Similarly, it is also known by the name of "Renault Group", in English

maximization based on the driver driving skills and habits, map directions depending on current traffic, among others. Alongside with the services we find the Advanced Driver-Assistance Systems (ADAS) functionalities that increase the benefits of connected vehicles at the expense of adding an additional layer of complexity to the vehicle's embedded software. These driving assistance systems are based on a scale of 0 to 5, where 0 represents the normal vehicle, isolated from other systems and devices with no driving automation, and 5 represents the fully autonomous driving car [2]. Such intelligent vehicle represents the last step of digital evolution within the automobile industry; an entirely connected vehicle able to manage both internal and external communication networks whose only one target is to achieve the highest value in comfort, kindness to the environment, and security.

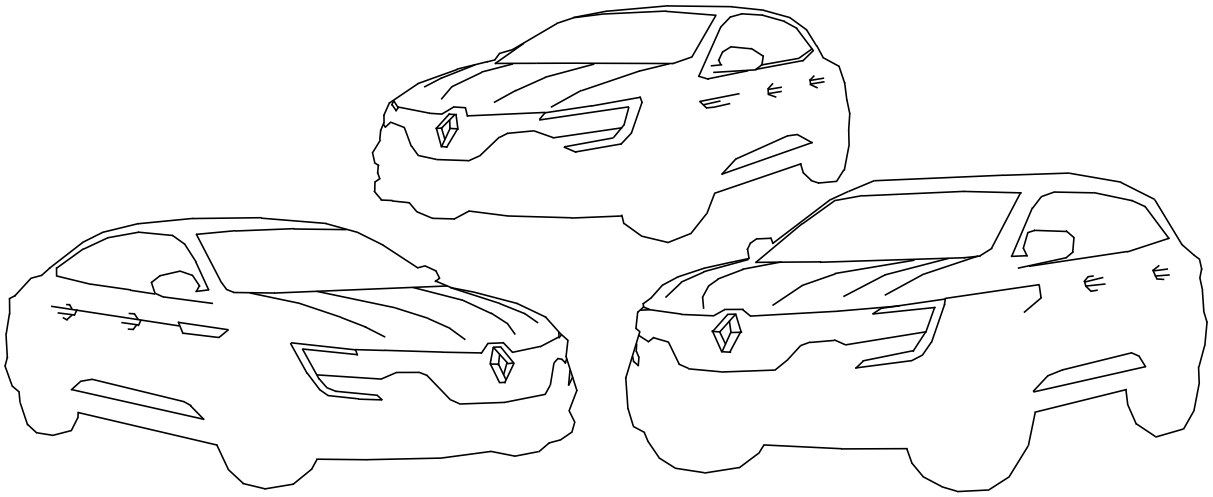


Fig. 1.1 Renault vehicles

One of the two halves that are key to achieving success in the car digital transformation is the inter-communication network. It lies between vehicles and environment devices and is directly based on the upcoming Fifth Generation (5G) technology. This 5G standard will bring a peak data rate of 20 Gbit/s, user experienced data rate of 100 Mbit/s, mobility of up to 500 km/h, 1 ms latency, and many further benefits [3]. The inter-communication network will allow efficient traffic management, drastically reducing traffic jams and deaths. Nearly 1.3 million people die in road crashes every year, and over 50 million result injured and/or disabled [4]. Around 90% of road accidents are attributable to driver error [5]. If the vehicle precisely knows the position of other vehicles at any moment under any circumstances, intelligent synchronization techniques shall apply and no more round-abouts, traffic-lights, or any other human coordination techniques are to be required. We know that car jams are caused by the human being and his inability to manage high density traffic situations, for they have slow reaction times and short attention spans. Therefore, self-driving cars are a structurally systematized solution that increase traffic throughput. Besides, other probable outcomes are to apply: no car blockages will result into smaller commuting times, augmenting the spreading surface of cities and allowing higher commuting distances, autonomous driving will spare the necessity of parking slots in the city centers, plus increasing the number of potential customers that do not have a driving license, just to name but a few. However, safety and the price of required investments are the main obstacles to the growth of such intelligent transportation systems [6]. It is clear that current roadside infrastructures are not ready to cope with the outstanding growth of connected vehicles. They are not ready to confront all required entities to carry out experimental testing. There is nowadays a lack of connected devices that allow data gathering and processing related to traffic, comfort and safety, enabling data sharing to transportation systems. Fortunately, those connected objects are soon to

come and they will bring connected vehicles into reality thanks to the Internet of Things (IoT). Thus, a smart grid framework will be implemented. As a matter of fact, connected technologies will result in structural changes in the automotive industry. They will modify the sales process by bundling connected services and features at the point of sale. As a result, they will account for at least a figure of 20% of automotive sales by 2025 [7].

The other half is the so-called intra-vehicle communication network, relying entirely into an internal structure, or Controller Area Network (CAN) network. It allows excellent synchronization between all available Engine Control Unit (ECU) at the vehicle's on-board computer and enables data sharing between all embedded systems, such as the camera, laser, lidar, and so forth. As a general idea, the multimedia car system is also connected to this network. It is composed of both the In-Vehicle Infotainment (IVI) and the In-Vehicle Communication (IVC); the IVI carries out the core functions, processing data and displaying it to the user, while the IVC is fundamentally the car's modem, allowing external connection and featuring some security functions. As an example of use, the IVC uses its Third Generation (3G) or Fourth Generation (4G) Radio Frequency (RF) antennas to establish a connection to the server, enabling a data exchange link that will be used by the IVI to request maps information to the server and display it to the user once it is downloaded by combining it to the IVC's Global Positioning System (GPS) signal. At both internal and external networks, the transmission of data and the data itself are a major and therefore must be secured and protected at any cost.

1.2 Project objectives

All in all, this document focuses on the simpler type of connected vehicle, the one that is currently under development and will be sent to production on late 2020 and released on 2021. As previously mentioned, there is a need of a smartcar that provides an answer to the current market requirements in terms of connectivity; a minimum number of remote services is mandatory. Based on the concern of data privacy as a result of the European General Data Protection Regulation (GDPR) law implemented on 25th May 2018, as well as the physical implications of having external data access to the vehicle's CAN network –the one that controls the camera and radar but also the engine and break, we must fully analyze, understand and safeguard the vehicle's cybersecurity. To do so, there is an exhaustive mechanism set in place within the Groupe Renault fully-owned Renault Software Labs (RSWL) company to study and write requirements based on said study, design, implement, and automate different Test Case (TC)[2] in order to validate akin complex embedded system, that will be wholly presented later on in this document.

More particularly and straightly applied to this project, I will design, implement and automate a variety of test cases aiming to partially validate the embedded software running on the vehicle's IVC from both a cybersecurity point of view and a performance perspective. Consequently, I will require a test bench, or set up, conforming the required environment that will allow me to initially carry out manual tests and eventually automate them via the Micro-services Automotive Test Robot Integration eXecutor (MATRIX), a RSWL proprietary tool that allows you to do so. Such environment will emulate the IVC's connection to the Base Station (BS) via the CMW500, a wideband radio communication tester

[2] I will use both "test case" and TC, its acronym, equally throughout the document

–or simply "comm tester"[3]. By using it, I will be able to modify physical parameters such as frequency, path attenuation, transmission power, to name but a few, at will. Besides, I will connect the comm tester to a real Internet connection so we can link the IVC to VNEXT, the Groupe Renault server providing remote services –further described on Chapter 2.2.3. The whole setup will serve as basis to deal with real systems in real circumstances. I will try to take the system[4] beyond its limits and, if possible, find possible bugs. To do that, I will use several computers to carry out simple attacks and sniff network data traffic. Thanks to the ulterior use of automated test cases, I will be ready to add another brick into the validation building that is the connected vehicle. Initial simple but powerful test cases related to cybersecurity will allow me to partially certify the vehicle's modem is protected. During this validation process, I will analyze the embedded software, and produce code that will be used in development centers belonging to the Groupe Renault located both in France and abroad.

Thus, the main objectives of this individual project are:

1. **Build a test environment** –test bench or setup, **acknowledge it, and automate it**. Further described in Chapter 4, it is composed of:
   (a) A CMW500, or wideband radio communication tester, emulating a network Base Station (BS).
   (b) An In-Vehicle Communication (IVC), the vehicle's modem providing network connection to the car.
   (c) An external PC, able to act as an external device within the same network than the vehicle.
   (d) A packet sniffer, located in the same network than the car, allowing me to capture any external digital communication.
2. **Analyze the external connectivity** of the system, composed of:
   (a) A connected car, through its modem, or IVC, that gives it such connectivity and connects it to the Groupe Renault server, called "VNEXT", and to the Internet.
   (b) A remote server, the so called VNEXT, allowing for remote services and functionalities. It is ready to accept incoming connections from any previously registered car's IVC.
3. **Analyze both the cybersecurity and performance of the IVC**
   (a) I will only cover the inter-vehicle network. I will consider the inner intra-vehicle network secure and thus protected.
   (b) I will develop simple manual testing via different tools, such as "OpenSSL", "Scapy", "Wireshark", and Python libraries for data science. They will be described later on.
   (c) Related to performance, I will be using different guidelines and requirements coming from the Power and Performance (PnP) team at RSWL.
   (d) Related to cybersecurity and as "Assistant Engineer in Cybersecurity", I will be using different guidelines and requirements coming from:

[3] CMW500, wideband radio communication tester, communication tester, or simply comm tester will be used as synonyms from now on

[4] Anywhere in the document, when I say "system" I mean both the IVC and VNEXT working together

i. The Groupe Renault cybersecurity experts located at Technocentre Renault (TCR), the main Research and development (R&D) center in Paris, France.

ii. Previous work from Xavier [8], a former RSWL cybersecurity intern whose work serves me as fundamental basis to this document.

iii. My research, work and findings.

4. **Automate both cybersecurity and performance test cases and integrate them into MATRIX** –as shown in Chapter 5, confirming it as the definitive automation tool within the Groupe Renault:

   (a) To date, all cybersecurity tests are manually conducted. Thus, full automation will allow building a more complex testing environment in which cybersecurity test will consequently increase in complexity. This project represents the initial step towards such purpose.

1.3 Project structure and schedule

This document[5] summarizes all work and development I conducted during my "Contrat de professionnalisation"[6] at RSWL, a 12-month contract allowing me to combine both the student and employee status[7] at the same time. It comes as a result from a collaboration agreement between RSWL and the Institut Supérieur de l'Aéronautique et de l'Espace, or Higher Institute of Aeronautics and Space, in English (ISAE-SUPAERO), my Engineering School in France. During my last year of studies, I was given the opportunity to benefit from two different calendars:

- **From September 2018 to March 2019**, I attended classes from Monday to Wednesday. I chose data science and Artificial Intelligence (AI) as my major, so I could learn about Machine Learning (ML) and Reinforcement Learning (ML) algorithms and techniques. In parallel, I was enrolled into a school project based on developing a Capture The Flag (CTF) in AI[8]. Both Thursdays and Fridays I attended RSWL to work on this Master's Thesis project.

- **From April 2019 to August 2019**, I fully attend[9] RSWL as a software developer. My official status is "Assistant Engineer in Cybersecurity", or "Apprenti Ingénieur" in French.

Choosing a software company specialized in embedded software to carry out my final engineering project was a thoughtful idea I considered carefully. As an international dual degree student attending two different engineering schools under a collaboration agreement, I wanted to work on a project that would jointly serve as basis for both institutions. The connected vehicle represents a complex system that brings two different concepts together. On the one hand we have the telecommunications

[5] Written in LaTeXso images can be zoom in without bearing bad resolution. Besides, any reference included in this document has an hyperlink to its source.

[6] Also known as "Contrat d'apprentissage" in French, but the latter usually lasts 3 years instead of 1. It is similar to an extended-in-time internship but bearing the dual status of student-employee

[7] Therefore this project is entirely based on product development and not in research.

[8] The website may be consulted at https://wait-ctf.com/

[9] To date, July 2019, I'm still working on the project

perspective. **Connected** means networks, protocols, and Radio Frequency (RF) communications. It perfectly suits the Escola Tècnica Superior d'Enginyeria de Telecomunicacions de Barcelona, or Barcelona School of Telecommunications Engineering, in English (ETSETB)[10], my Engineering School in Spain where I attend the "Master's Degree in Telecommunications Engineering (MSc)" program. On the other hand we have the transportation perspective. **Vehicle** means mobility and embedded software. They are two of the most important topics covered at ISAE-SUPAERO, my Engineering School in France where I attend the "Ingénieur ISAE-SUPAERO (MSc)" program. It is well known that nowadays transportation systems rely entirely on intelligent embedded software that efficiently manages all different scenarios. This Dual Degree Master's Thesis represents the ultimate stage of an international collaborative engineering program allowing students to learn and love engineering from two different but complementary points of view.

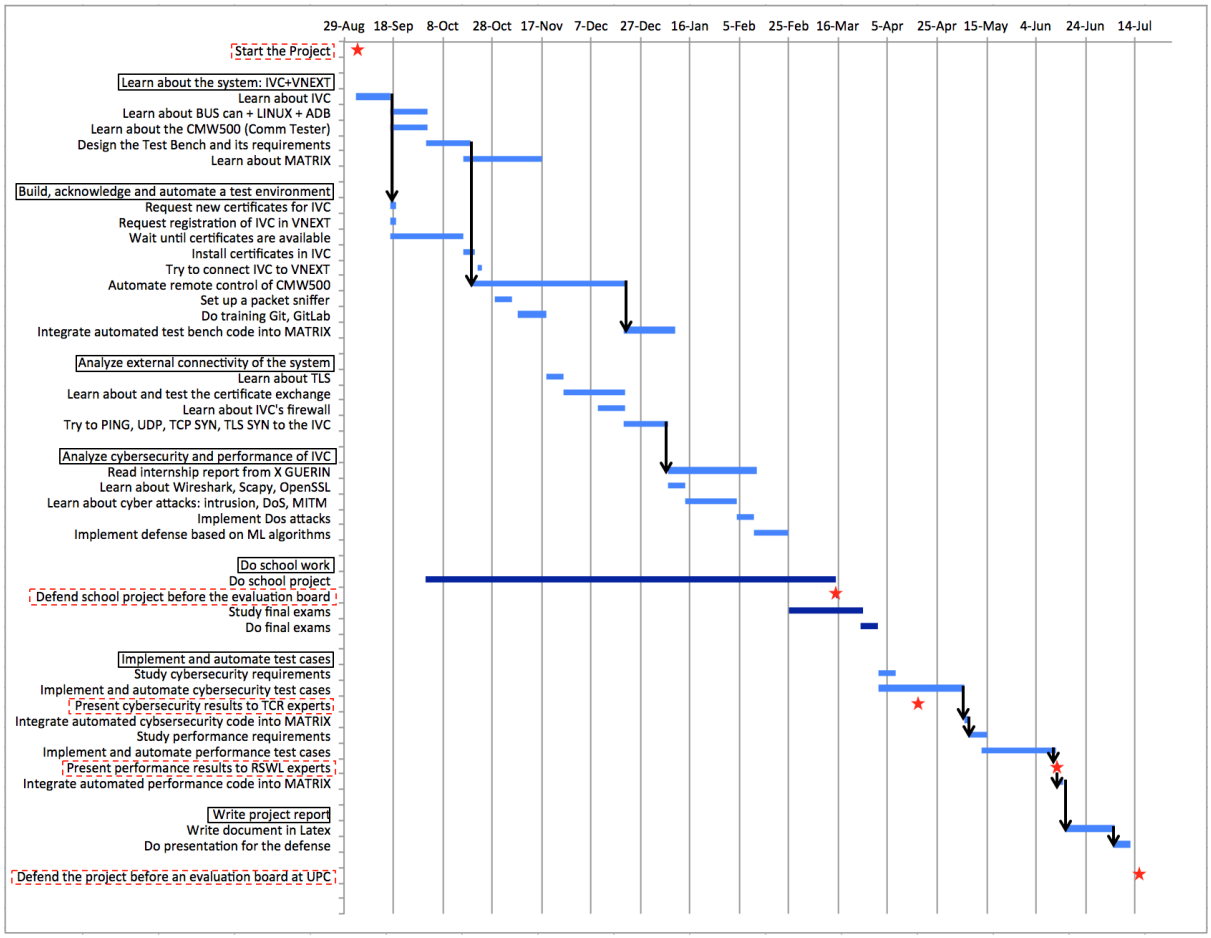


Fig. 1.2 Gantt Chart of the project

This project has 4 main milestones –signaled as a red star in Figure 1.2. Basically, the start and the end of the project, as well as the presentation of my obtained results before the experts of the Groupe Renault. Additionally I decided to include an external milestone referring to the school project that I attended at the same time than this project as complementary information –in dark blue. Besides, the Gantt Chart shows how I made slow progress during the first half of the calendar, for I was attending RSWL only two days a week.

[10] School belonging to the Universitat Politècnica de Catalunya, or Technical University of Catalonia, in English (UPC)

Chapter 2

Renault and connected vehicles

2.1 About Renault

2.1.1 The Groupe Renault and the Alliance

The Groupe Renault, a carmaker founded in 1898, is an international multi-brand group that brings together the Renault, Dacia, Renault Samsung Motors (RSM), Alpine and LADA lines. It is present in 134 countries and sold over 3,9 million vehicles in 2018. Besides, its revenue in 2017 was 58,7k M€, with a net income of 5,2k M€ and a workforce of over 181k employees. To date, it is the world's leading French vehicle-manufacturer [9]. Groupe Renault means "one group, five brands". More precisely:

Fig. 2.1 One group, five brands

- **Renault**, with over 2,6M vehicles sold in 2017, is leader of the European electric vehicle market.
- **Dacia**, with over 655k vehicles sold in 2017, it offers a wide range of simple and reliable vehicles at affordable prices.
- **Renault Samsung Motors (RSM)**, with almost 100k vehicles sold in 2017, is one of the top five carmakers in South Korea, particularly reputed for its service quality.
- **Lada**, with over 335k vehicles sold in 2017, became a Groupe Renault brand in January 2017. It is the long-standing leader of the Russian market.

- **Alpine**, founded in 1955, was back to production in 2017 with the new Alpine *A*110 reflecting lightweight, compactness, and agility.

The Groupe Renault main targets are (1) becoming the leader in electric vehicles with over 8 electric models and 12 electrified models[1], and (2) achieving 100% connected vehicles in key markets and 15 autonomous Renault services. Besides, 21 new models will be released in the upcoming 5 years –from 2017 to 2022. Some further information released on January 2019 [10]:

- In Europe, registrations were stable (+0.5%) in a market that grew by 0.2%. The Group's growth comes mainly from the B segment (Clio, Captur, Sandero), and New Duster. Clio remains the second best-selling vehicle in Europe and Captur the first crossover in its class.
- In Renault brand's electric vehicle segment, sales increased by 37% over the year, with an acceleration in the second half (+62%). Renault is the European leader with a 22% market share.

However, the Groupe Renault does not walk alone. It constitutes one of the three entities composing the Renault-Nissan-Mitsubishi (RNM) Alliance[2]. Such Alliance is the most sustainable and productive multi-cultural strategic collaboration in the global automobile industry. It is in fact the world's leading automotive alliance. Founded in 1999, it offers an unique, pragmatic, agile model of design and production, being always ready to evolve and integrate new projects and partners.

The Alliance sold over 10,76 M vehicles in 2018 in 200 markets worldwide, becoming the world's top 1 car seller in the world[3]. Some keys [11]:

- Over 10,76 M vehicles sold in 2018, meaning 1 in 9 vehicles sold worldwide
- Over 775k electric vehicles sold
- Over 450k employees worldwide
- 122 manufacturing plants

Among the multiple benefits from the RNM Alliance, we highlight the following:

1. 80% of the Groupe Renault produced on shared Common Module Family (CMF) platforms.
2. 4,2k M€ in savings thanks to common at both investments and R&D levels.
3. 18 k M€ invested in R&D by the Groupe Renault, giving access to a portfolio of technologies worth 50k M€.

The Alliance targets zero-emission, connected, autonomous, affordable mobility. In order to achieve those goals, the Groupe Renault acquired Renault Software Labs (RSWL) in 2017.

[1]Electric models are the reason number 1 behind the Fiat Chrysler Automobiles (FCA) merge proposal to Renault in 2019. Besides, factories remain little automated.

[2]Renault owns 43,4% of Nissan. Nissan owns 15% of Renault. Nissan owns 34% of Mitsubishi Motors. The Alliance board is equally composed at 50-50 by Renault and Nissan.

[3](1) RNM Alliance: 10,76 M, (2) Volkswagen Group: 10,62 M, (3) Toyota: 10,39 M, (4) General Motors: 8,38 M, (5) Hyundai-KIA: 7,29, (6) Ford: 5,98 M, (7) Honda: 5,25 M, (8) Fiat-Chrysler: 4,84 M, (9) PSA: 3,88 M, (10) Suzuki: 3,33 M.

2.1.2 Renault Software Labs

Renault Software Labs (RSWL) was born in 2017. The Groupe Renault acquired Intel's French RD activities specializing in embedded software, making it a fully-owned subsidiary. It employs over 450 distinguished engineers holding extensive international experience in software development. It is located on 2 different locations: Toulouse and Sophia-Antipolis. It belonged from 2009 to 2017 to Intel's Mobile Communication Group (MCG), a specialised software testing branch of Intel. Such acquisition was a decisive movement, a strategy implemented targeting excellence in terms of connected and autonomous vehicle development.

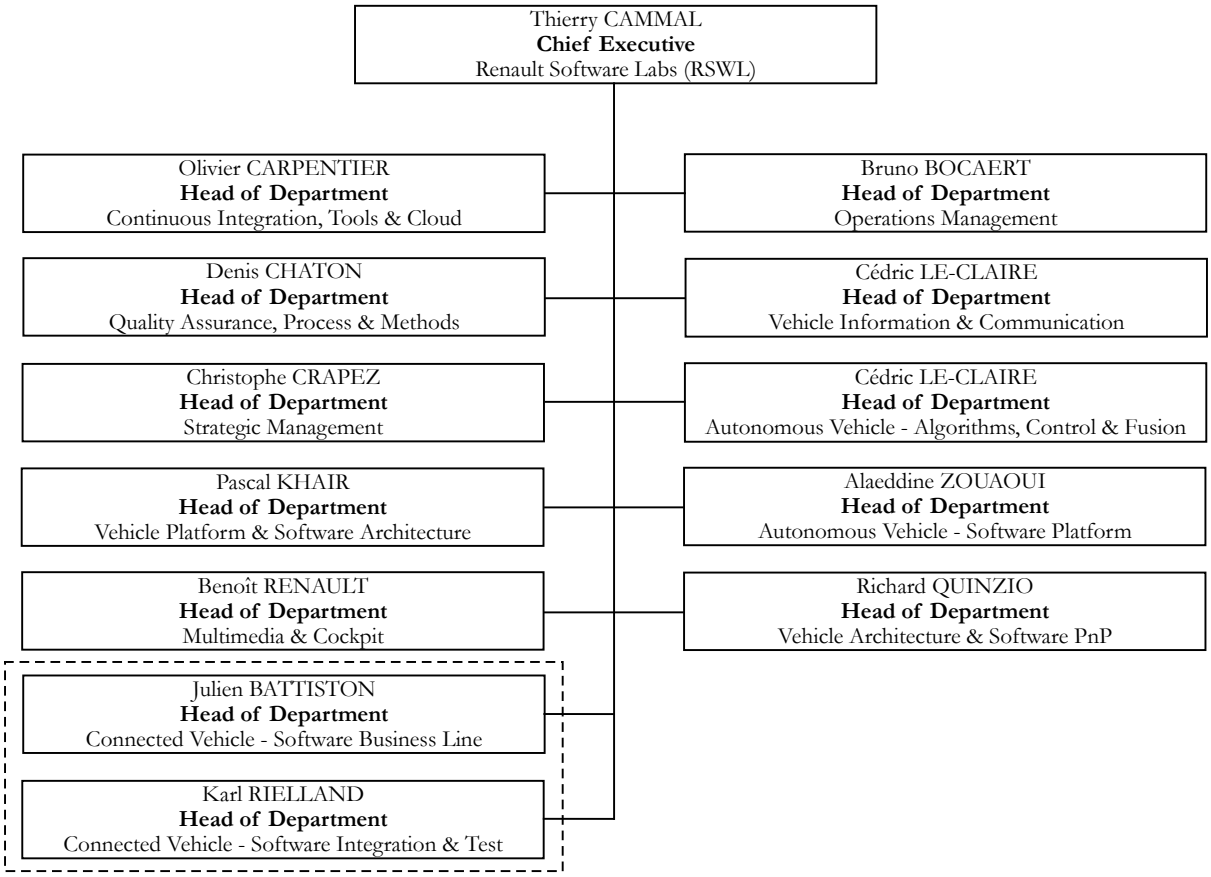


Fig. 2.2 organization chart of RSWL

As can be seen from the Figure 2.2, RSWL is composed of 12 different departments; the 7 on the left are located on Toulouse premises, and the other 5 on the right located on Sophia-Antipolis premises. As of January 2018, there were 9 different projects or divisions; this document is entirely based on the CCAR project, the so called software line –highlighted in Figure 2.2 by a dashed box. The Software Business Line (SBL) department focuses on the connected car development from a business point of view, cost analysis, project viability, and so forth, while the Software Integration and Test (SIT) department focuses on the connected car development from a pure engineering perspective based on software coding and hardware at a laboratory scale.

Connected Car

The so called Connected Car (CCAR), main project which I belong, lines up all roadmaps and milestones towards the creation of added value within RSWL. It includes all the software tools that enable remote development and integration at a global scale, focusing on what brings value to the company: the low cost vehicle. Its main purpose is to create such a tool that is capable of providing full traceability and continuous software integration on a weekly basis. In other words, it shall assure the internationalization of code development on a regular basis. At an Alliance scale, there is a complete pipeline of stakeholders where the CCAR represents the last, but not least, part of the sequence. As any other business, it all starts from the market demands. They are then translated into features assigned to different squads. After validation, those features are designated to the platform teams that will develop and consequently deliver solutions based on continuous integration. It allows adding new features weekly and add value accordingly. After final End-to-end (E2E) validation, those features are ready to be included into the vehicle's catalog that will form the different vehicle programs.

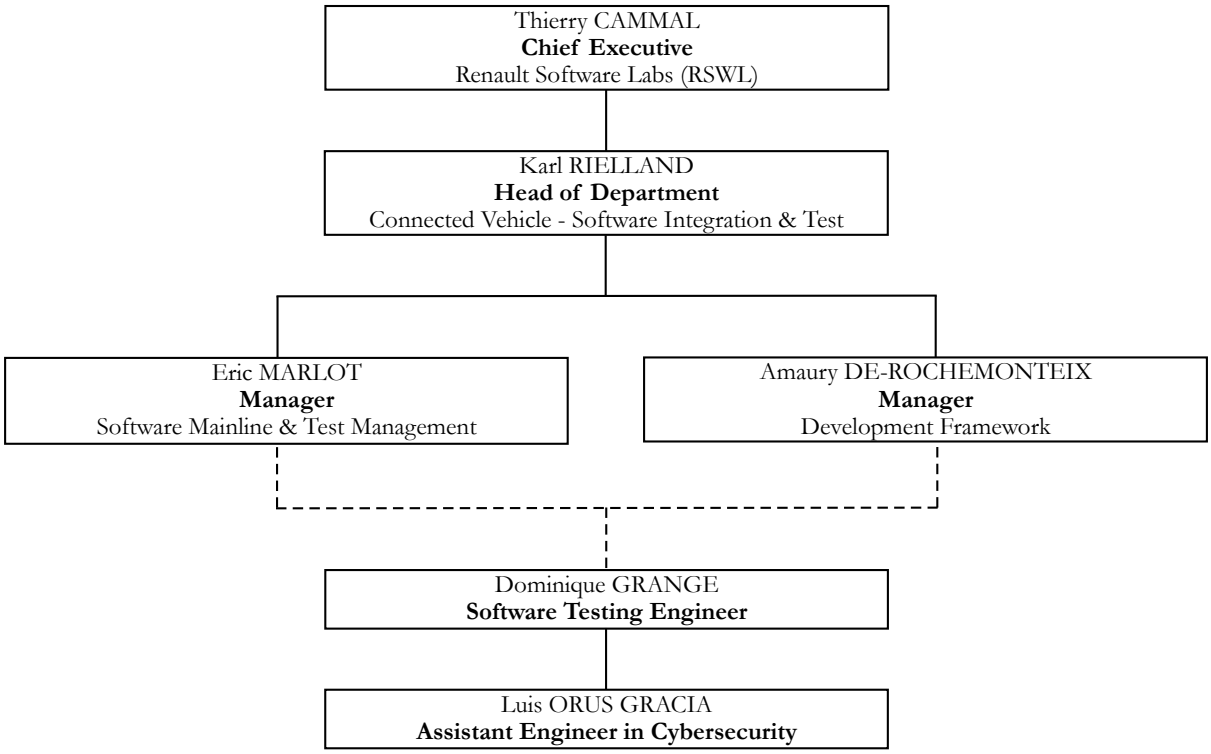


Fig. 2.3 Organization chart of CCAR which I belong

2.2 Fundamentals of connected vehicles

As stated in the Oxford Dictionary [12], connected means "brought together or into contact so that a real or notional link is established". Thus, connected vehicle means data exchange, or remote contact, between the vehicle and any other system, such as on-board sensors, roadside infrastructure, cloud services, other vehicles, and so forth, in order to increase safety, efficiency, and User Experience (UX). Even though stand-alone internal services and applications such as amazing sound systems or high-definition screens –see Appendix A, are vital into high-end cars, connectivity is key. Paradise

remains in the outside world; the extraneous environment lying just one click away. Thus, any current demanding vehicle must bear a wireless communication. Cars are becoming less transportation and more entertainment and comfort, both ideas meaning the Internet. In order to keep drawing the attention of customers, car makers must step forward and bring new functionalities that upscale the user experience.

When we say connectivity, we refer to wireless mobile telecommunications technologies such as 3G, 4G, and soon-to-come 5G, Wireless Personal Area Networks (WPAN) technologies such as Bluetooth, IEEE 802.11 WiFi protocols, and location technologies such as GPS, GLONASS and Galileo. In terms of added value, navigation is essential. Not only do we mean the underlying RF technology, or the communications standard, but also the upcoming new features. These are third parties Internet services within the vehicle's ecosystem, secured in-car WiFi hotspot, smart antennas, and the promising soon-to-be 5G. They will all require modifications on the vehicle's architecture, at both software and hardware levels; to completely redesign the platform [13]. As previously stated, such connectivity is achieved by means of two main components: the In-Vehicle Communication (IVC), or vehicle's modem bearing both a RF and a antennas, and the In-Vehicle Infotainment (IVI), an Android running on top of a Linux Operating System (OS). Both IVC and IVI are manufactured by the so called "Tier 1" –an external manufacturer providing such components. For example, the IVC used in this project was supplied by Continental, a German automotive manufacturing company.

2.2.1 Certificate enrollment

At the Groupe Renault, any connection or data through to the network is protected via the Public Key Infrastructure (PKI)[4], a cryptographic system. Based on the International Telecommunication Union (ITU) X.509 public key certificates standard and, mainly, the Rivest-Shamir-Adleman (RSA) asymmetric encryption cryptosystem. Such asymmetric cryptography uses complex mathematical operations and algorithms[5] to produce an unequivocal key pair assigned to every involved party –i.e. the sender and the receiver. While in conversation, a sender can use its private key along with the message to *sign*[6] it and ensure only legit communications and proper authentication. Analogously, a sender can use the receiver's public key to encrypt a message and send it through the communications channel. The recipient, only, will then use its private key to decrypt it. X.509 digital certificates are an electronic document used to prove the ownership of a public key –or key authentication. We rely the inter-communications security to a hierarchical architecture relying on a trusted third party –the "root" Certificate Authority (CA). It is an entity that issues legit digital certificates –a "proof of authenticity". A digital certificate is like an Identity Document (ID). It is built from a combination of different variables associated to an entity, such as name, organization, validity period, to name but a few. By taking all previous information, we can use the private key to compute a hash[7] function –i.e. the Secure Hash Algorithm (SHA) function, and then obtain a digital signature. Again, it is a mathematical scheme allowing for verifying the authenticity of a digital document. Any involved party

[4] Architecture based on a unique pair of public –disseminated widely, and private keys –kept secret and known exclusively to the key owner.

[5] Mostly based on the factorization of large prime numbers.

[6] Signing creates a "digital signature". It is a mathematical function ensuring data integrity, meaning no data was modified during the communication.

[7] One-way mathematical function that maps arbitrary-size inputs into fixed-size outputs.

will subsequently add its own certificate to the message, building the so called "chain of certificates". For example, if A wants to send a message to B through C, A will form a digital certificate, append it to the message, and send it to B. Accordingly, B will apply the same procedure and create another digital certificate, append it to the message and the previously added A's certificate. C will then receive a message plus two digital certificates issued by a trusted third-party CA. Thanks to the root CA, C is now sure it was indeed A and B, and no other external parties trying to spoof the system, the ones that send and transferred such message.

As it will be later on presented, the Groupe Renault relies on Transport Layer Security (TLS), a very well-known secure transmission protocol –further presented in Appendix B. Crucial information such as certificate exchange, supported cryptography algorithms, ciphers, and further configuration values are shared during the TLS session establishment. The whole system is based on trust. Therefore, certificates must be entirely unique and secure so no fraud may be found. Certificate enrollment implies all previous required steps in order to ensure legit certificates at both sides, from the vehicle's IVC to VNEXT, the Groupe Renault server. VNEXT acts as the root CA, enabling secure registering process and providing Firmware Over-The-Air (FOTA), remote services, and more. Without an audited certificate enrollment process, no secure connections can be ever guaranteed. Basically, the Renault's PKI goes as follows:

1. The root CA –VNEXT, sends a valid, unique certificate to the supplier –or Tier 1.
2. The Tier 1 then manufactures an IVC, and generates a pair of keys –public and private. It then stores them into the IVC, sending a Certificate Signing Request (CSR)[8] to VNEXT –server accounted for registering all IVC. It then sends the IVC to the car factory to be introduced into the vehicle.
3. When received in factory, such IVC is then assigned to an unique Vehicle Identification Number (VIN). This information is transmitted to VNEXT: both software and hardware are now bound.
4. After having received the CSR, VNEXT signs the IVC's digital certificate as root. It then sends it to the vehicle.
5. Once the mutual authentication is successful, they can communicate to each other in a secure way –by using TLS.

2.2.2 SOTA/FOTA

Even though we may install the latest functional software, it is doomed to become outdated. It is just a matter of time. Not only do new features and functionalities require a constant update, but also working software may drive into problems if the communications ending gets modified. This inevitably leads to one of the things the customer fear the most: the planned, or built-in, obsolescence. More precisely, limited useful life where devices or systems stop working properly and current functionalities are no longer available. In response, automotive engineers must use the so called software "over-the-air" to always provide ready-to-use software that is continuously updated over the car's lifetime. It is not a

[8] Containing the public key for which the certificate should be issued, identifying information, and the digital signature.

new technology, for we have been using it every time we updated the software of our smartphone for updating applications or registering for new online services. More precisely, we use FOTA to remotely update the firmware, the underlying software –thus Software Over-The-Air (SOTA), that allows low-level control of specific hardware such as radar, lidar, brakes, cameras, to name but a few. It also enables the possibility of double check. That is, obtaining an inventory of currently installed software. FOTA requires a working Subscriber Identification Module (SIM) card that identifies and authenticates the subscriber enabling mobile connectivity. A valid SIM card has many different elements that makes it unique: Integrated Circuit Card IDentifier (ICCID), International Mobile Subscriber Identity (IMSI), Personal Identification Number (PIN), Personal Unblocking Key (PUK), and a variety of information related to security authentication, ciphering, temporary data about the local network, and all the available services.

One of the simplest but at the same time very powerful SIM card functionality is that of awaking the vehicle. While not in use, the vehicle's on-board computer is on battery-saving mode. Upon arrival of , the whole system leaves the previous mode and activates all systems facilitating full connectivity. This automatically enables remote services, later introduced at present Chapter. FOTA allows the vehicle to connect to a server full of contents; it is seen as a centralised system where all important decisions and processes are managed on the cloud. They allow the so called "update campaigns" where several patches and enhancements are released and customers are requested to apply and install those changes by previously downloading it. FOTA services are key to the Groupe Renault, enabling three decisive remote functionalities. These are:

- **Update**, allowing customers to obtain the latest released working software product version that may solve previous bugs and malfunctionalities. It is directly downloaded from the remote server's repository.
- **Setting**, allowing the remote server to enable/disable different available services and benefits based upon subscription; some of them may only be applied upon a previous payment.
- **Inventory**, allowing the remote server to obtain the vehicle's information related to software and hardware and compare it to the repository.

Recently, Tesla announced an update campaign where some already available functionalities would require a new payment subscription [14]. It means that connected vehicles usually come with all necessary hardware to run brand-new upgraded software. However, car manufacturers use SOTA/FOTA to obtain all information related to the vehicle's inventory, allowing the user to download new updates but only enabling those settings if certain requirements were previously met. Remote software may be triggered after three different reasons: (1) the remote server –VNEXT in Renault, triggers it after a recent update, (2) in a regular basis previously established by the FOTA team on the development centers, and (3) on demand. It would be risky otherwise, for external connectivity may be only activated if requested by the user. From a cybersecurity point of view, we must avoid unnecessary connections and reject all input traffic if it was never asked for before. It means that the server will never send, by its own, unsolicited messages –same goes for you banking provider who will never require you to transfer your banking credentials by email. Thus, both the settings and inventory functions will only ever occur if the customer previously agreed to carry out a software update. All three functionalities

"update", "setting", and "inventory" share similar communication protocols. The whole system uses TLS, a very-well known protocol designed to provide privacy and data integrity between two or more communicating computer applications. To summarize:

- **Download**. The vehicle informs the user about new content available. Once the user agrees to download it, the vehicle's communication system will start a secure connection to the server's repository. It will then download the software package.
- **Authentication**. Thanks to the use of a chain of certificates, there is a valid mechanism to unequivocally verify that only trusted parties were involved in the communication process. The vehicle is programmed to only accept data directly coming from the server, by verifying its identity using digital signatures (non-repudiation).
- **Integrity**. Thanks to the use of hash functions, we can ensure that the message was not modified as it travelled for an unprotected communication channel, susceptible to Man-In-The-Middle (MITM) attacks.

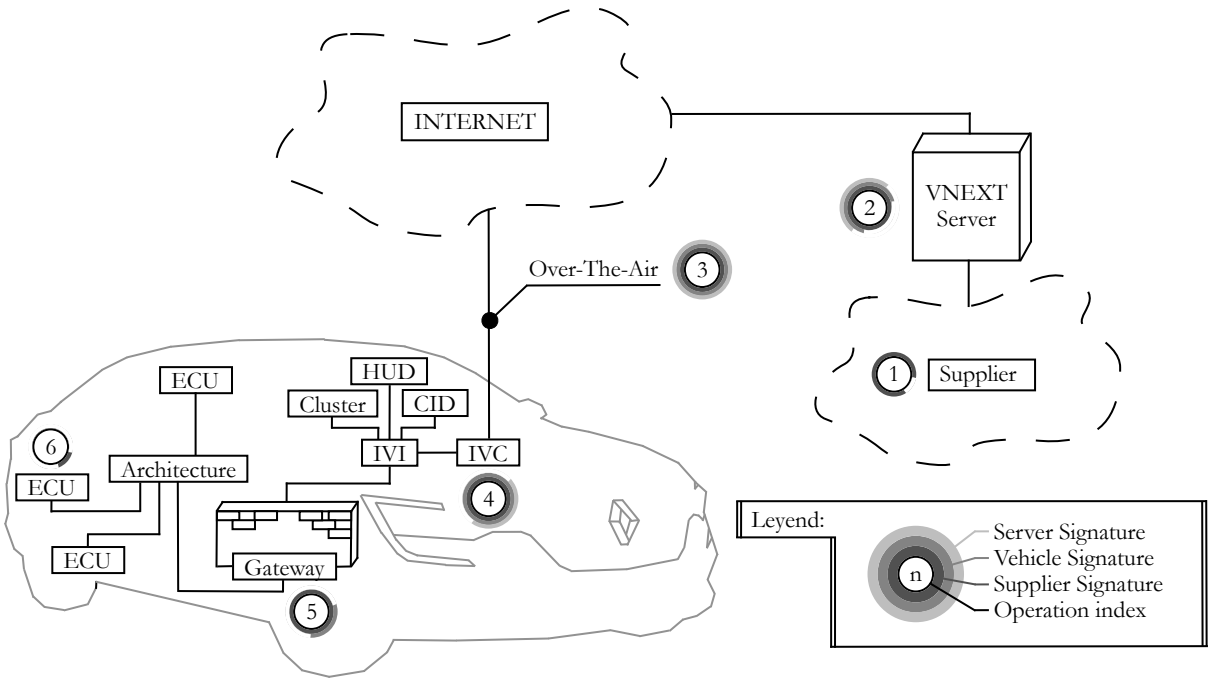


Fig. 2.4 Firmware Over-The-Air

Following the Figure 2.4, we can highlight the following points during a package download:

1. The supplier delivers a functional code package thanks to the use of CI, as will be presented on Chapter x. It then signs it with its private key so its authenticity is assured, and applies a hash function to the package so its integrity can be verified later on.
2. The server, namely VNEXT, stores every update and runs every remote service. It manages the "software campaigns". If requested and initialised by the client, or the vehicle's IVC, it will establish a TLS connection. See Annex X for further details with regard to the Public-key cryptography, and an example of a typical TLS connection.

3. If both communication entities are successfully authenticated, a FOTA over a secure TLS connection over the Internet is carried out.

4. The package is successfully downloaded at the client's side thanks to the use of the IVC's modem. The IVC will then pass the package over the IVI, the intelligence bearer. As seen, relevant information may be shown at the displays, all of them managed by the IVI.

5. Once the gateway, or vehicle's architecture inner firewall, is successfully transversed, the package is ready to be transferred via the architecture towards the target ECU.

6. Once every verification process is done including the integrity of the package, it is then installed on the target ECU and finally activated. For example, a new software improving the current Emergency Brake Assist (EBA) system is released. Therefore, new software will be installed on both Brake Control Module (BCM) and Suspension Control Module (SCM) ECUs. It is a very sensible process given that many important components may be affected. Malfunctioning software present at the BCM may drive into a fatal accident.

Even though we are talking about the same technology and way of thinking, we must be clear on this matter: the lifecycle of cars is different to the consumer world. We tend to change our smartphone every 24 months or even less, but we tend to keep our cars for a longer period of time. In 2016, the average age of passenger cars in the EU was 10,7 years; in France, it was 8,8 years in 2017. According to ACEA, "the EU motor vehicle fleet is getting older year-on-year" [15].

2.2.3 Remote services

Remote services are used to provide a safer, funnier, more efficient customer experience. They supply some advantageous functions such as connected navigation, remote access to information, and convenient assistance services. They give truthful meaning to the name "connected" during the first initial phase of connected vehicles. They only require a secure connection to enable feature-extended capabilities. Therefore no intelligent, connected infrastructure is required. At an Alliance level, more than 90% of all vehicles are expected to be connected by 2022 [16]. They mean an added value to a vehicle experience[9]. From both marketing and strategic point of views, remote services are key to bring added value to the driving industry. Some examples of remote services are:

1. Remote Keyless System (RKS), or **remote lock and unlock**, allowing the user to use its phone application to authenticate himself and ask for the remote unlock of the vehicle. This way, there is no need to have a mechanical key that is likely to get lost. Besides, no proximity is required, allowing the use of this remote service anywhere –where a network connection is available, anytime.

2. A variant remote service of RKS the so called Remote Keyless Ignition System (RKIS), allowing for **remote start of the vehicle**. It may be convenient when in extreme weather conditions. For example, if in very cold weather and low temperatures, you may start your engine early in the morning before going to work in order to warm up the engine and turn on the A/C[10] heat.

[9] Mostly remote services are based on proprietary paid phone applications such as "MyRenault App".
[10] Air Conditioner, climate control, or climatization.

Alternatively, when in very warm weather, you may turn on the A/C prior to getting into the car to have a smooth and pleasant temperature when entering the vehicle later on. Along with the RKS, it opens a wider range of applications. For example, an owner can lend his vehicle to anyone anytime. They may also facilitate the process of renting a vehicle since only a remote access through the internet would be required to both unlock the vehicle and start the engine. Besides, it may also boost the car sharing paradigm.

3. Remote Horn and Lights (RHL), allowing for a wide range of applications from finding the vehicle in a crowded parking by **emitting both visual and sound effects** or illuminating your path when in low sight conditions.

4. Further remote services account for **remote GPS location** –knowing the exact location of your vehicle at anytime, allowing for optimal traffic management or anti-theaft applications, or **remote battery management** allowing for battery optimization, to name but a few. Future autonomous vehicles will bring remote parking, light-less autonomous driving, and so forth.

2.2.4 Google Automotive Services

"In September 2018, the Alliance signed a global multi-year agreement to partner with Google to equip Renault, Nissan and Mitsubishi Motors vehicles with intelligent infotainment systems. It will be based on Android, the world's most popular OS, to offer customers a new array of services including Google Maps, the Google Assistant and the Google Play Store, scheduled to start in 2021" [16]. More precisely and from an engineering point of view, we introduce the Google Automotive Services (GAS).

GAS is a multi Original Equipment Manufacturer (OEM) compatible ecosystem allowing full development of automotive on-board applications benefiting from previously agreed mobile standards featuring diverse modular solutions targeting different markets [17]. It provides full stack Android's software development and integration, enabling Hardware Abstraction Layer (HAL) adaptation to the Linux Kernel lying beneath. It brings integration between the Android Open Source Project (AOSP) and the different mobile services, running on the car's IVI.

One of the main concerns on the automotive industry is that of "why spending tons of money on infotainment systems when there is a quicker and simpler solution being simply using your own phone to achieve such functionalities?". It is a fact. Everyone now holds their phones to the right of their steering wheel and continues to use the services their smartphone provides and which they are used to. It is hard, if not nearly impossible, to convince a faithful user to use a different platform. That is why GAS offers a unique, continuous UX designed to reinforce the idea of brand experience and keep the phone-to-car –therefore phone-to-IVI, transition as smooth and simple as possible.

2.2.5 Vulnerabilities: towards cybersecurity analysis

Again, this project aims to implement different automated tests in order to verify that the is a secure system from a network point of view –see inter-vehicle communications. In other words, there is

a need to run some tests in order to certify that the IVC is not "easily hackable"[11]. As engineers, we work together side by side to build a robust system requiring several development steps. In terms of cybersecurity vulnerabilities, one simple idea is key: there is a need to limit the risks, rather than reduce the threats. Connected vehicles are such complex systems and therefore vulnerable –the average modern high-end car software is 100 million lines of code, to be compared with 39.5 million lines in 2009 for Windows 7 or 13.8 million lines for a Boeing 787 [18]. We find two different cybersecurity attacks: those who seek to gather data in order to sell it or use it perversely –i.e. user location tracking attacks [19], and those who seek to harm the system, resulting in malfunction, interoperability, or even full damage. Several standards are highly recommended when applying the "security by design" paradigm [20]. Generally speaking, a car is a heavy, dangerous object. If "hacked", it can become a serious risk to the physical security of people. Thus, cybersecurity focuses on protecting its hardware, firmware, and software, ensuring only legit communications, detecting abnormal events or anomalies, and managing its lifecycle.

We will assume that development tools, methods and processes –such as policies, procedures and guidelines, where chosen with cybersecurity in mind so systems are secure by design. Many standards, guides, and/or good-practices manuals such the ones from the ECS or ETSI have been released in order to boost cybersecurity at a company basis. ISO 27000 family of standards that "helps organizations keep information assets secure" [21]. ISO 26262 applies for electric and electronic systems within the automobile industry [22]. We will also assume that all required cybersecurity concerns have been applied on the Linux Kernel. Besides, cryptographic tools like hash and encryption function are also considered to be secure. This means that possible failures will be based on misconfigurations, but not on protocol or underlying technology malfunctioning.

Connected vehicles are linked to both internal and external communication networks. This necessarily means that there are many different vulnerable points of entry. Among all of them, we highlight three large groups:

- **Mobile connectivity**: an access to the global Internet Protocol (IP) network via RF antennas where both FOTA updates and remote services lie. They require a secure layer to protect the user data privacy. They encompass any information related to real people, such as location, behavioral patterns, banking information, and so forth. To ensure only legitimate communications, we use secure protocols like TLS. This project focuses on this group.
- **WPAN**: composed by both Bluetooth and WiFi technologies. For example, the new Bluetooth version 5.0 enables extended range up to 400 meters and faster data transfer.
- **Physical connectors**: any physically accessible interface such as the USB port.

It is said that 62% of security experts expect hackers will start using AI [23]. Thus, there is a need for finding advanced threats. For that, we need to **visualize**, by using behavior analytics and forensic trails, **understand**, by automation and making machines learn up to x60 faster, and **stop**, by applying dynamic policies, layered security, and detection-identification-response automation. Thanks to the use of AI, we can learn through machine and deep learning from unprecedented, new attacks. We can

[11] Meaning it is not either externally accessible nor its communications compromised.

reason by gathering insights and extract data patterns. We can also augment and respond to threats faster. Ideally, we will implement cognitive computing defenses that are adaptative, interactive, stateful and contextual.

During this document I will implement several accessibility and connectivity tests in order to verify if such mobile connectivity is secure. I will eventually use some AI techniques to visualize data and eventually come up with a basic intrusion detection algorithm.

Chapter 3

Validation and testing

Before we tackle the topic of validation, we must provide some useful definitions to lay the ground for. Integration is a "testing performed to expose defects in the interfaces and in the interaction between systems". Verification is the "confirmation by examination and through the provision of objective evidence that specified requirements have been fulfilled". Validation is the "confirmation by examination and through provision of objective evidence that the requirements for a specific intended use or application have been fulfilled" [13].

3.1 Continuous Integration

Continuous Integration (CI) is a development practice in which any developer belonging to a project upload –called check-in or push, his/her code to a shared repository on an iterative basis several times a day. Once the code arrives to the common archive, a pipeline, automated build, or set of required test, is launched. It will fail if any integration error or malfunctioning is detected during the pipeline process. It allows teams to detect problems early. CI major outcomes are faster code delivery, faster and easier debugging, quicker integration times, easier integration processes, higher visibility level and better intra-team communication. CI means ensuring no new patches can lead to failures, or bugs. The system must behave, at least, as exactly as before; a new functionality will be added, but no security breaches or new bugs shall ever appear. After a proper upload, developers may request to integrate their code –called merge request, from their own development copy version of the project –called branch, into the project main repository –called master branch. Repositories are based on "Git", an open source distributed Version Control Systems (VCS) [24]. Once the developer pushed the code, made a merge request, and successfully integrated it, we say a patch was applied. CI good-practices go as follows: (1) maintain a single source repository, (2) automate the build, (3) make your build self-testing, (4) everyone commits to the master branch, (5) every commit should build on an integration machine, (6) fix broken builds immediately, (7) keep the build fast, (8) test in a clone of the production environment, (9) make getting the latest executable easy, (10) make the whole process visible so anyone can see any previous change, and finally, (11) automate deployment [25]. In plain English, you initially download the whole project and therefore create a copy to which you apply your modifications safely. Once you believe your code is good to go, you upload it to the main repository.

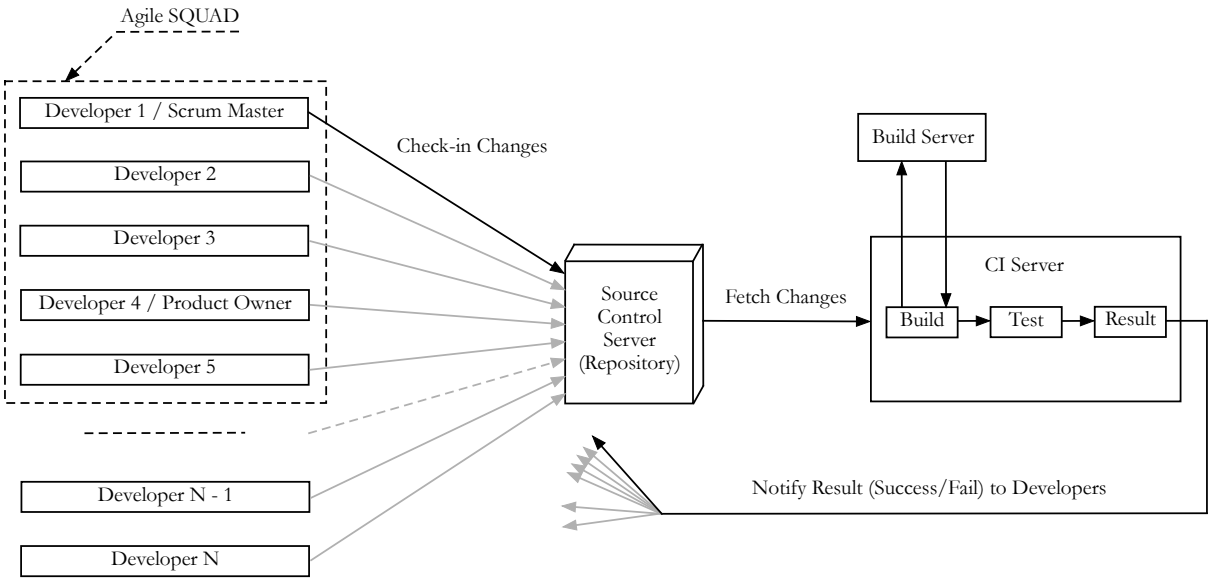


Fig. 3.1 Continuous Integration

As also shown in Figure 3.1, the find the "Agile SQUAD", just another name for "Agile Development Team". It accounts for the "Agile software development" paradigm, mostly based on self-organizing and cross-functional teams [26]. They answer to "don't just fix the product, fix the process too" [27]. Essentially, the companies who did not fix the process such as BlackBerry[1] or Motorola[2] resulted into failure. On the contrary, Netflix, Spotify or Apple succeeded to face scalability as they grew. More precisely, RSWL applies the "Scrum" framework [30]. Basically it promotes and favors courage, focus, commitment, respect, and openness of software developers within a team. A minimum size scrum is composed by a Product Owner (PO), the developer team –composed by a front office developer and a back office developer minimum, a Quality Assurance (QA), and, last but not least, the "Scrum Master". While the PO focuses on maximizing the value of the products created on the squad –or development team, the Scrum Master ensures that everyone understands the Scrum process, being responsible for promoting, guiding and supporting it. For example, RSWL SIT implements scrums composed of a 3-week sprint. It means all the work previously included within the initial backlog and planning must be finished when the sprint is up. All year long calendar is divided into many sprints forming groups taking 3 months in average. A whole year is divided into 4 slices, or quarters (Q) –therefore a year has around 16 sprints, or 4 Q of about 4 sprints each. There are daily scrum meetings of about 15 minutes where all scrum members report a little bit of what they did yesterday and what they will develop today, at a very high-level without going to much into detail. This way all squad members have a general idea about the whole product development process.

Belonging to the world of "continuous" we find two additional terms: "continuous deployment", or "continuous release" where every good build is released to users, and "continuous delivery", making your software always ready to be delivered, causing the process to be business-driven and not IT-driven. "While continuous deployment implies continuous delivery the converse is not true" [31]. Deployment requires a fully automated process as a means of smoothly transferring the software to the user.

[1]BlackBerry mobile phone market share: 9% in 2007, 19% in 2009, 4% in 2012, negligible share today [28].
[2]Motorola mobile phone market share: 23,5% in 1997, 14,3% in 2007, 1,9% in 2012, negligible share today [29].

3.2 Validation process

RSWL software test development is entirely based on ISO 29119, an “internationally agreed set of standards for software testing that can be used within any software development life cycle and by any organisation”. It is currently based on a series of five published standards: 29119-1:2013 for concepts and definitions, 29119-2:2013 for test processes, 29119-3:2013 for test documentation, 29119-4:2015 for test techniques, and 29119-5:2016 for keyword driven testing [32].

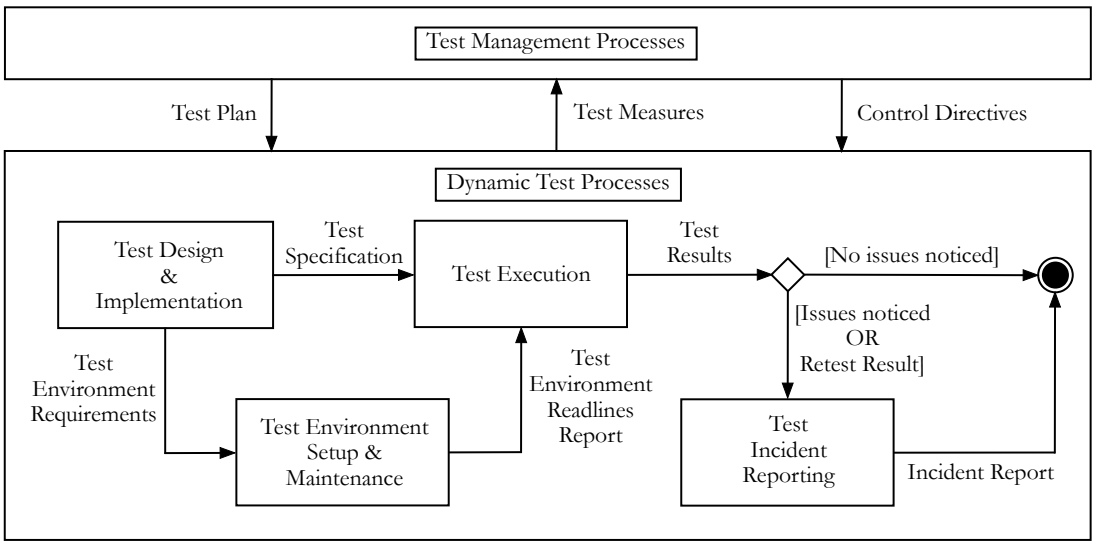


Fig. 3.2 Basic scheme of operation of ISO 29119 [13]

As shown in Figure 3.2, it is based on two, well-differentiated domains: Test Management Processes and Dynamic Test Processes. The whole system requires a test strategy that integrates the needs coming from both the test projects and test operation teams. The Test Design is in charge of delivering a TC catalog and all requirements for Test Environment, being the latter responsible for test automation and test bench definition. Designing a system is a very complex task. It requires full planning in order to wholy achieve a system that is well designed, “secure by design”, fully functional, effective and efficient. In order to do that, a comprehensive study is carried out so all available options are taken into account. Once the architecture is defined, it then starts a development process resulting into a final validation process through complete testing. It is here where the Test Design block appears. It is a fundamental entity fully in charge of defining all requirements related to the system and, based on them, create a test catalog that includes all tests that can be actually executed, are under current development, or are to be implemented. In plain English, it is a database of what tests can be done, what was previously done, and will be eventually done.

A TC, or again "test case" is “a set of preconditions, inputs, actions (where applicable), expected results and post conditions, developed based on test conditions”, where a test condition is “an aspect of the test basis that is relevant in order to achieve specific test objectives” [34]. A test case verdict may be PASS, where expected result of all actions are “pass” meaning they went as intended by meeting the requirements, BLOCKED, where expected result of one precondition or pots condition is “fail” meaning they did not go as intended based on the requirements, or FAIL, where expected result of an action not being pre or post condition is “fail”. During this project we will only present results being either PASS or FAIL. Test cases may be executed manually or automatically. We say that an

action plus its expected result is a "test step" if executed manually, belonging to a so called "shared step library" within the manual world. A test case may be composed of one or several test steps. If executed automatically, we then refer to it as the "keyword library", resulting in "keyword driven testing" within the automated world. Finally, a generic set of test cases is defined as "test catalog". Test execution is built from actionable test cases. The test catalog at RSWL, contracted as SWCAT [13], is Xray, a complete test management tool for Jira. As defined on its website, Jira is "the 1 software development tool used by agile teams". It is a Java-written proprietary issue tracking software developed by Atlassian, an Australian software company founded in Sidney.

It is here where we find the main function of the Software Mainline and Test Management (SMTM) team, based on the "Test Design" and "Test Catalog". It lays the groundwork for enabling such system. According to their own principles, its main mission is to "develop and maintain test coverage for the integration and the qualification of RSWL software projects". They must successfully manage diversity, meaning many different products –or systems, for example several different IVI enabling different capabilities, are present and therefore some steps need to be coded differently based on its architecture and platform and ideally highly adaptable. Subsequently, we find the Development Framework (DF) team incharge of both test environment and test automation and whose roots are based on the Continuous Integration (CI) paradigm, further described in the upcoming section.

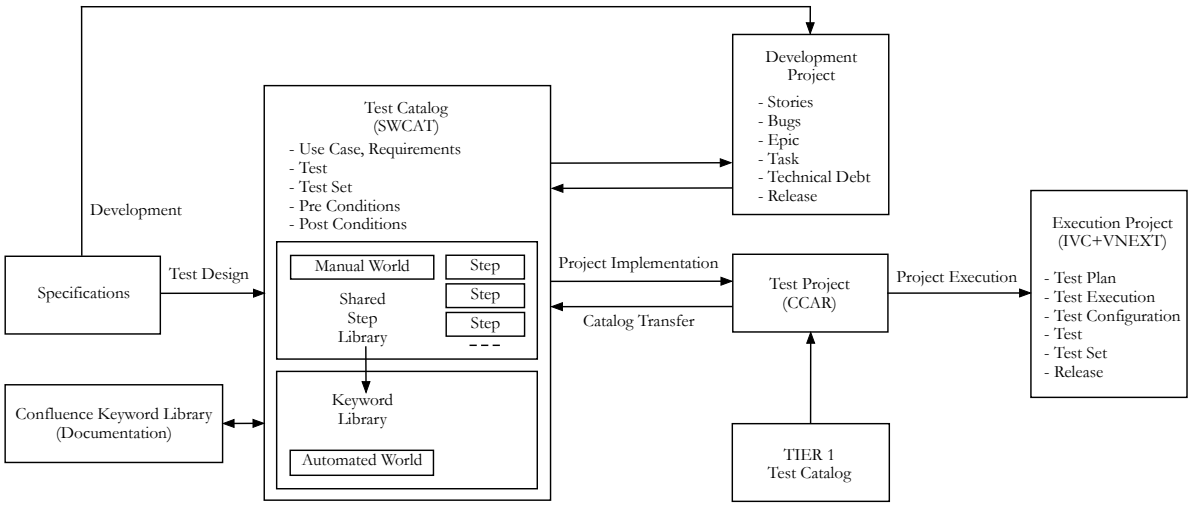


Fig. 3.3 Test Catalog: a complex process

The whole process is shown in Figure 3.3. It also includes "Confluence", the RSWL Wiki, being "a powerful collaborative tool to communicate between coworkers to quickly share knowledge to everyone's benefit". It provides a simple but complete tool that enables you to easily write and publish a wiki document to the Renault's intranet at the "renault.com" domain. Several people can collaborate to the same project –and thus same wiki page, letting viewers to correct wrong already published information. In this particular case, we use Confluence as keyword library to find customized step procedures per vehicle and release configuration.

3.3 Test framework: MATRIX

Micro-services Automotive Test Robot Integration eXecutor (MATRIX) states for Micro-services Automotive Test Robot Integration eXecutor, and it is "the automation environment defined by RSWL CCAR SIT for integration, regression and validation activities across the Alliance" [13]. It is similar to Docker. It allows people who either do not master or even do not know how to code in Python to design and implement test cases. It provides an abstraction layer by hiding the working details to the non-programmer coworkers.

MATRIX is based on Robot framework, a "generic open source automation framework for acceptance testing, acceptance test driven development (ATDD), and robotic process automation (RPA)" [33]. It provides full source code availability, an online, always-ready, supportive development community, and entire development independence. Besides, it is not clear that open source development frameworks are weaker than proprietary solutions from a cybersecurity point of view. Both WannaCry and NetPetya viruses infected proprietary code [20]. The contrary is not verifiable either though.

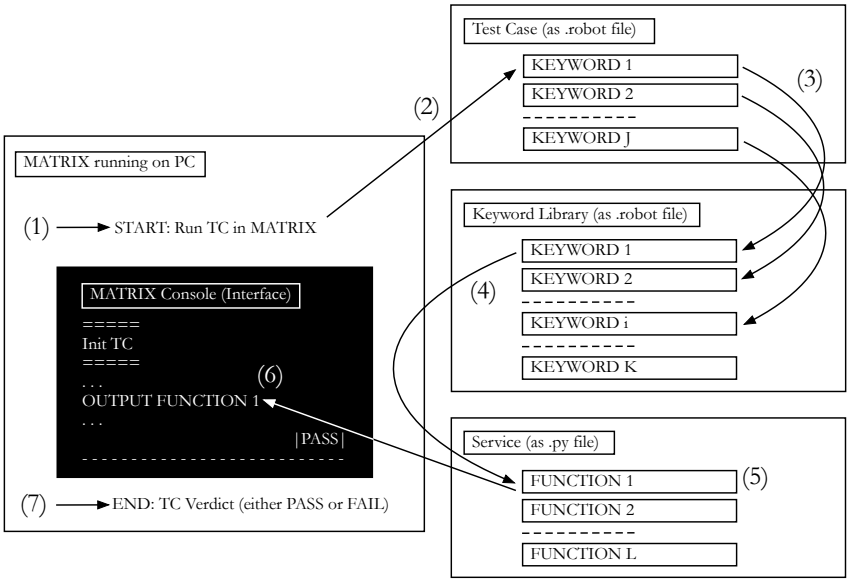


Fig. 3.4 TC execution in MATRIX

Basically, Figure 3.4 explains how a test case, or TC, is automatically executed step by step in MATRIX. It expresses in a clearer way the concept previously introduced in Figure 3.3. Concretely, let's say we want to run an automated TC in MATRIX (1). We will suppose that all keywords –name for automated steps, have been previously coded allowing modular programming. Therefore, we will build our TC taking into account our TC requirements –normaly provided to us by the test design team. To do so, MATRIX will then execute (2) all keywords that we had previously introduced into our TC (3), from 1 to J. They all belong to a keyword library of size K with $K >= J$. There, every available keyword is linked to one or more Python methods included within a Python class within a service –here having L different methods. Usually one service is assigned to one MATRIX developer –I will be the developer in charge of the comm tester service as well as the cybersecurity service. Any chosen keyword (3) from the keyword library (4) will then run a specific Python method (5) –i.e. keyword K is never executed in Figure 3.4 as it was not previously included on out TC. As shown, such Python method, if requested, will print information into the MATRIX black console (6) –or to a

log file, allowing for debugging. Normally, the Python methods are coded to return either TRUE or FALSE based on the initial requirements –meaning they are PASS or FAIL. If one or more methods –and therefore keywords, return FAIL, then the result of the TC is FAIL. On the contraty, if all methods work as intended and are PASS, then the result of the TC is PASS (7).

For example, let's say we create a TC based on 2 different keywords. Both are linked to a single Python method. The first method verifies if a PING get an answer from our PC to the Internet. The second method verifies if a DNS request gets an answer to a valid domain name on the Internet. If both methods verify such connectivity and therefore return TRUE, our keywords will be PASS and, as no FAIL are present, the TC verdict will be also PASS –in green. Thus, having a TC FAIL means one or more keywords were not successful meaning the expected outcomes were not met.

3.4 Cybersecurity validation, a key to success

Connected means data exchange. Exchange means interaction between two or more parties, risking of appearance of unwanted third-parties. Data means information. Information is knowledge, and knowledge is power. Here lies the importance of cybersecurity in general, and this project in particular. There is today a growing concern about connected vehicles at both cybersecurity and data privacy levels; a clear perception of lack of security. In France, 48% of people claim they never share personal information online, versus a 19% in the US [35]. They are top priorities for customers and should be even of higher importance for automobile manufacturer companies. Even though cybersecurity has gained momentum lately, it is still at the starting point. Companies worldwide are becoming aware of the importance of protecting their systems. They are exponentially increasing their investments in computer security, computer forensics, and data analysis for intrusion detection. Returning to the automotive industry, it is believed that the global market for connected cars will grow by 270% by 2022, bringing new technology advances like digital key granting/revocation for Mobility-as-a-Service (MaaS) car-sharing, automatic car personalization and 3rd party access for courier and e-commerce solutions –all managed through your smartphone. However, they come along with a huge cybersecurity risk for brands and consumers meaning a security first approach for connected car apps and IVI is fundamental [36]. Therefore, as vehicles become digitalised and fully connected, the subject of security is becoming critical. Again, both cybersecurity and privacy concerns represent the biggest obstacle to the growth of connected cars for the 34% of all experts consulted, right before the difficulties related to technological capabilities, at 19%, and safety concerns, at 18% [7]. When we speak of "cybersecurity", we refer to the protection of hardware, firmware and software, ensure legitimate communications, detect unusual events or anomalies, and manage the life cycle of systems.

At an engineering scale, measures have to be taken. What was coded yesterday is being embedded today and will be used tomorrow. It means there is an imperative commitment related to what we previously created, and we must act accordingly. In plain English, if we detect tomorrow a coding error it will be too late. Thus, many corroboration steps are mandatory so there is a redundancy in terms of acceptance and certainty. By working in this iterative approach, we benefit from both validation and traceability outcomes. As a matter of fact, the first objective is always to achieve a 100% safety. The perfect mark where no errors ever appeared, do not appear, and will not appear. However, as engineers, we must come back down to Earth and face reality: failures are likely to happen and will

happen. Quoting Murphy's law, "whatever can go wrong, will go wrong". Thus, in the case of an improbable failure, the second objective would be to find exactly where the deficiency was. If it were to happen despite of its associated tiny probability, we need to be able to go back in time even more than 10 years of validation records. There's a fundamental need of precisely locate any possible failure and the reasons which caused it; many people's lives are at stake. Validation through automated testing and full traceability are the keys to success if the automotive industry is to exist in a digital world.

But, what is validation? What is testing? Literally speaking, what "validation through automated testing" means? Even though we will precisely cover all these matters throughout this document, a starting point needs to be established. According to the Oxford Dictionary [12], validation is "the action of checking or proving the validity or accuracy of something". Similarly, a test is "a procedure intended to establish the quality, performance, or reliability of something, especially before it is taken into widespread use". And last, but not least, automation means "the use or introduction of automatic equipment in a manufacturing or other process or facility", where automatic stands for "working by itself with little or no direct human control". Thus, we understand this process as a very intelligent way towards achieving our main purpose: come intelligent vehicles alive. In this matter, cybersecurity must be implemented in a bottom-up approach. In order to succeed in achieving a completely secure ADAS 5 vehicle, we must initially protect the ADAS 1. It's worth noting that the earlier we start thinking this way and planning accordingly, the better the upcoming autonomous vehicle will be. Build up the finest intelligent car, but vulnerable, would mean putting the cart before the horse. We shall start by applying simple but compelling tests, and set up a more powerful structure from it. If such connected vehicles are to exist, we must prove they are both legit and accurate, and demonstrate they can actively work under any circumstances.

Part II

Achievements

Chapter 4

Test bench for connected vehicles

4.1 Test bench

Like any other validation project, we need to set up a test bench so we can assess the system we are working with. Ideally, it must be precisely detailed and easy to reproduce, so we can expect the exact same outputs when introducing specific, case-dependant inputs. More precisely, according to the Oxford dictionary, we use the test bench to carry out a bench test, "a test carried out on a machine, a component, or software before it is released for use, to ensure that it works properly".

We find three different important entities within the test bench:

- **Network Base Station (BS)**, emulated by a Wideband Radio Communication Tester[1], or CMW500. It runs a set of applications on top of a Microsoft Windows XP:
  - **Data Application Unit (DAU)**, connected to a RJ-45 port located at the back of the device. It has an internal server. It enables network connectivity, allowing performance analysis.
  - **Signalling Application**, connected to the RF ports located at the front of the device. It provides full connectivity with Global System for Mobile Communications (GSM) (2G), Wideband Code Division Multiple Access (WCDMA) (3G) and Long-Term Evolution (LTE) (3.95G, being almost 4G) mobile communications standards.
- Connected vehicle, composed by the **In-Vehicle Communication (IVC)**, CAN bus, and power supply emulating the battery. The IVC requires two different antennas to work:
  - An **RF antenna** allowing network connectivity.
  - A **GPS antenna** allowing positioning and time alignment.
- **Remote server**, namely VNEXT, the Renault's database hosting services and certificates.

[1] Again, we name it "comm tester" for simplification purposes.

Thanks to the use of such test bench, we will be able to measure and modify:

- IVC power consumption, connectivity, firewall, and more.
- Several connection status, from cell enabled/Packet Switched (PS) connection established in telephony, network access to external websites, time to connection, certificate exchange, and more.
- Network parameters such as the underlying network technology, band, and channel, external attenuation, noise, to name but a few.

4.2 Scheme diagram

We will consider two different data links: (1) the **IVC-to-CMW500** link, meaning from the car's modem to the broadband cellular network technology via 3G, and (2) the **CMW500-to-VNEXT** link, meaning from the emulated BS to the Renualt's server via the Internet. At an Alliance level, such database for hosting services and certificates is called AVNEXT –where "A" stands for Alliance.

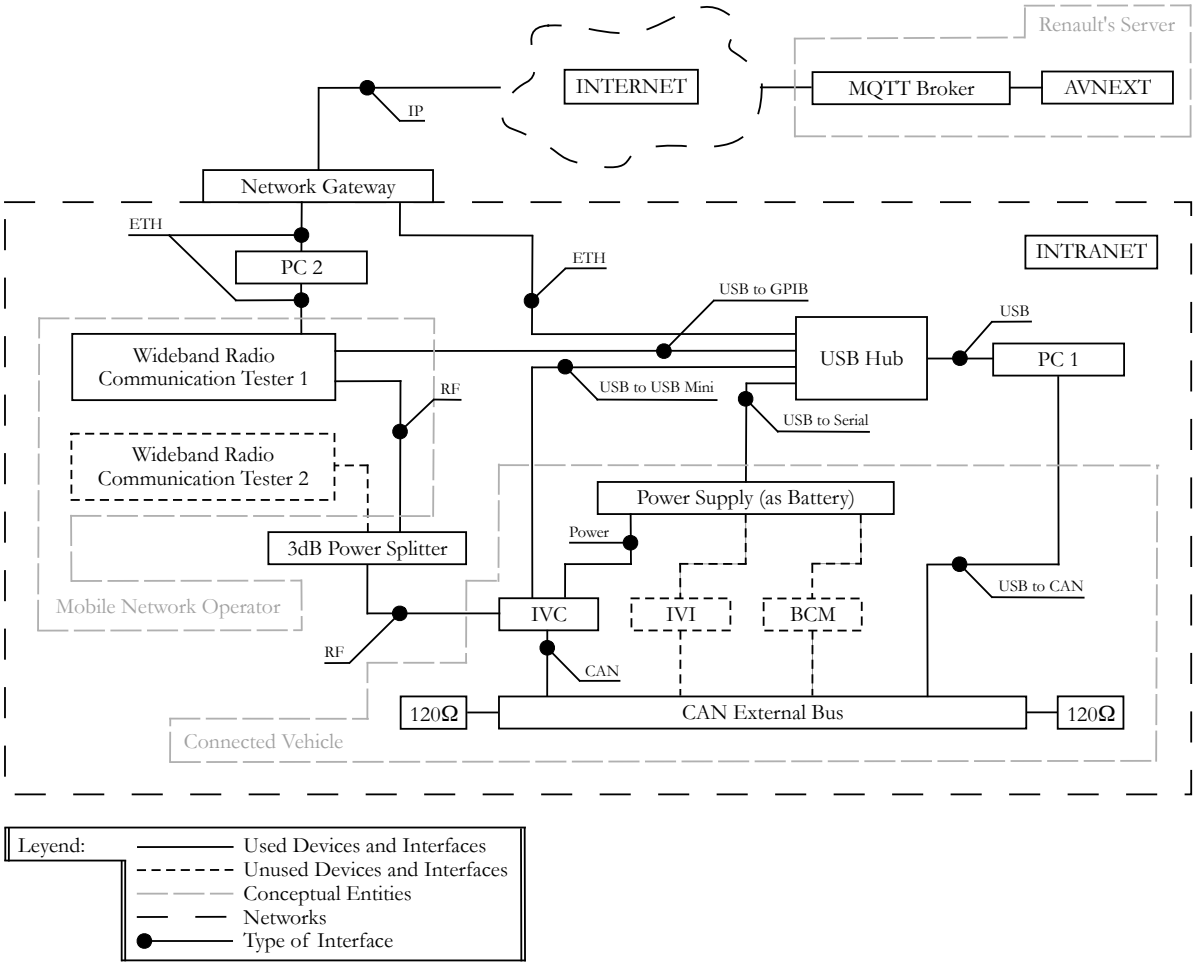


Fig. 4.1 Test bench

It was Dominique who settled up the "Connected Vehicle" part as shown in Figure 4.1 –a colorful version from it can be found on Appendix C.1. It means I initially received a working CAN bus and a functional IVC powered by a power supply, but containing no certificates. I was in full charge of settling up all the data connections by enabling the Internet connectivity through the comm tester –meaning making the system entirely functional. Besides, the detailed specifications of the involved entities and standards may be also found on Appendix C.2.

4.3 Configuration

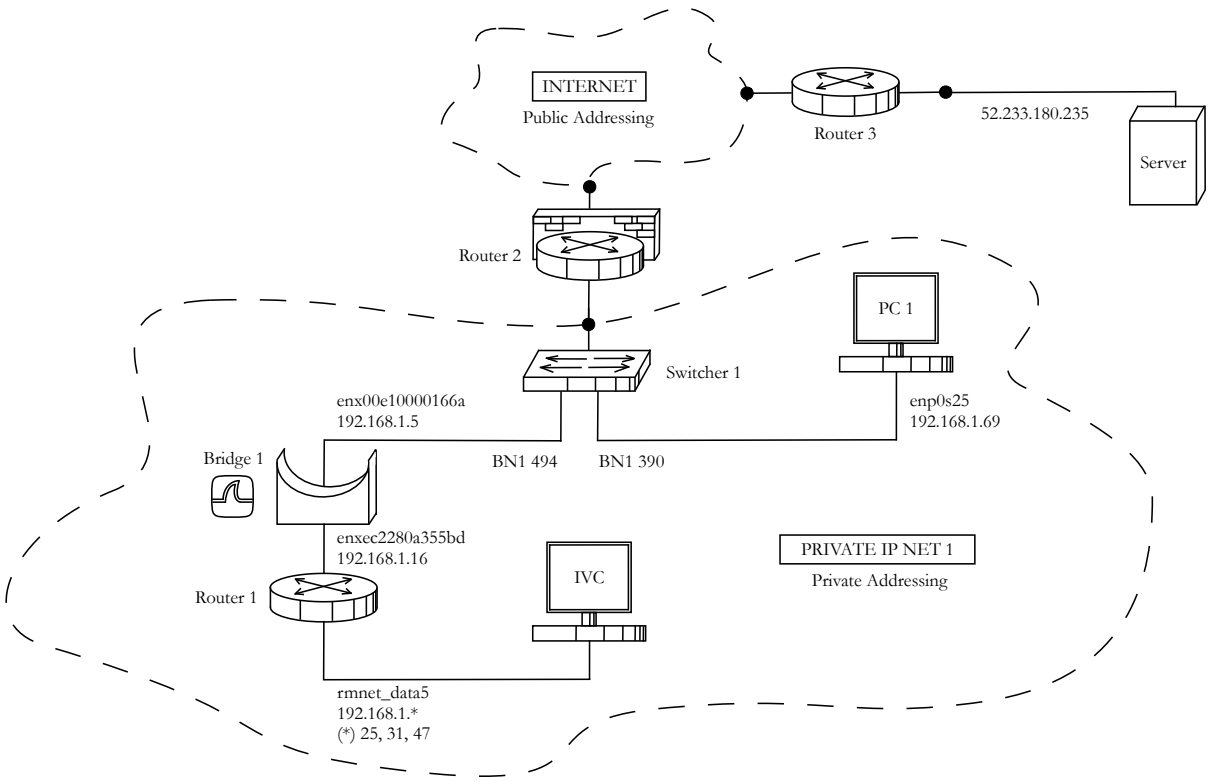


Fig. 4.2 Test bench at a network level

Several steps were required to make the system fully functional. These network configuration steps are, by chronological order of implementation:

1. Communication Tester –"Router" 1 in the Figure 4.2

   (a) Connect the device's LAN port (called "LAN DAU") to a in-wall physical socket, or more precisely into a RJ-45 Ethernet port, via an Ethernet cable, previously allocated with a static, private IP address (192.168.1.5) registered on the network switch (see "Switch 1") that is located in the building. The device is now behind the RSWL Internet's firewall.

   (b) Configure the Intranet's firewall. For that, the RSWL IT team must allow all incoming TCP and UDP traffic to the device. The device is now connected to the Internet.

   (c) Bypass the device's LAN Application to the Signalling Application (called "LAN REMOTE" and "LAN SWITCH") by using a short Ethernet cable. Both ports located at

the back of the device, as shown in the Figure 4.3. By doing this, we extend Internet connectivity to any device connected to the front RF connector.

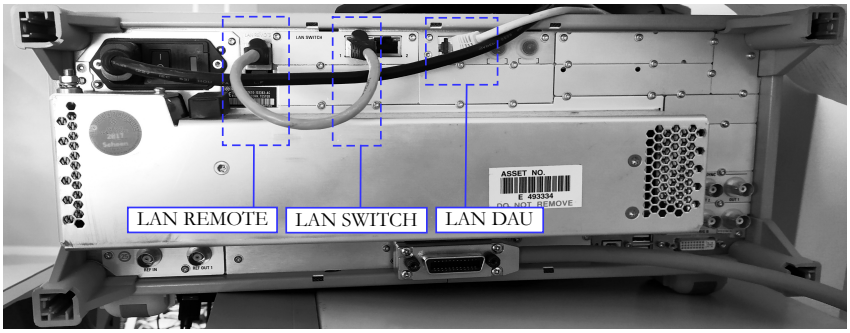


Fig. 4.3 Comm tester rear panel

(d) The device's internal DNS will provide private IP addressing to all devices connected to the front RF connector automatically –in other words, to provide Internet connectivity to all devices beyond the comm tester such as the IVC. It will use 192.168.1.16 as its own static, private IP address within the RF interface.

(e) All settings related to internal Applications (DAU and Signalling mainly), DNS, modulations, frequencies, powers, and so forth, are stored in the file "settings.yml". When properly loaded, it allows for automated settings installation. We will eventually install the GPIB drivers in order to remotely control the device.

2. IVC

(a) First and foremost, register the IVC in VNEXT. It requires a long and somehow complex procedure that may take up to one month to complete. Basically:

i. Contact the VNEXT certificate provisioning team located at the TCR and ask them for a new certificate based on the Public-key cryptography paradigm. It is generated from 3 different numbers related to the IVC (VIN, or Vehicle Identification Number, SN, or Serial Number, and PN, or Part Number). Once the certificate is created and stored on VNEXT, the server, we say the IVC has been "registered" into the system and the certificate is ready to use.

ii. Push the certificate and the pair of keys (public and private) into the IVC via the ADB commands. An IVC has an underlying Linux core running Android on top.

iii. Update the IVC's settings related to VNEXT, such as domain name and port number. The device is now ready to connect VNEXT –if and only if there is a working Internet connection.

(b) Light up the power supply at 12 V and initialise the CAN bus by continuously and non-stop sending CAN messages –we only deal here with "wake-up" and "normal use" messages. Without battery or any car commands, the IVC will remain inoperative.

(c) Verify via ADB that the IVC is online –meaning typing "adb shell" in a command window in PC 1 controlling the IVC via the USB-to-USB mini cable. A welcome message will be shown as in Figure 4.4.

(d) Connect the device to the comm tester via an RF cable. We will be using WCDMA, air interface standard found in 3G mobile telecommunication networks. The IVC we are

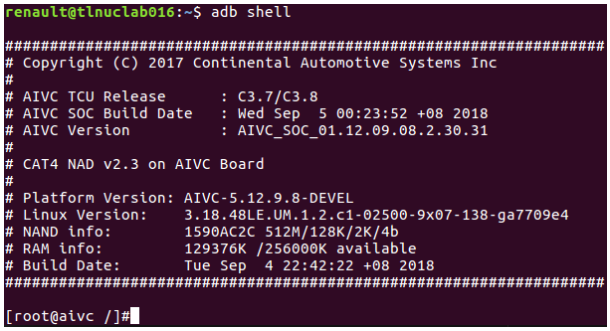


Fig. 4.4 IVC's welcoming message via "adb shell"

UTRA Band	Frequency (MHz)	Uplink (MHz)	Downlink (MHz)
1	2100	1920 – 1980	2110 – 2170
3	1800	1710 – 1785	1805 – 1880
8	900	880 – 915	925 – 960

Table 4.1 IVC capabilities: supported UTRA bands

working with is a device that only supports 3 different UMTS frequency bands, as shown in Table 4.1.

(e) The comm tester will automatically assign a private IP address to the device, being 192.168.1.∗ in Figure 4.2 with a ∗ of 25, 31, or 47 depending on the day.

(f) The IVC will then try to automatically connect to VNEXT via the Internet. If unsuccessful, it will keep trying based on an incrementational try-and-wait timeout.

3. Packet sniffer ("Bridge 1")

(a) Place PC 2 (from Figure (testbench)) between the comm tester (interface "ensex...") and the in-wall RJ-45 socket (interface "enx...").

(b) By running a simple Python code, we will bridge the two different interfaces. It will therefore forward all traffic coming from one to another and all the way around. We will call it "mybridge3".

(c) Start capturing traffic from Wireshark, a free and open-source packet analyzer.

4. Once points 1,2 and 3 are completed, the main router (we name it here "Router 2" for simplification purposes) will discover, if necessary, any MAC address located within the network and obtain its IP address via the ARP protocol. The IVC will then belong to another network located beyond the comm tester, thus behind the 192.168.1.16 IP address. The network setup will be therefore ready and functional. We have settled down and starting point.

4.4 Verification

Once the test bench is ready, including all required hardware and network connections, I need to establish a mechanism for acknowledgement. It means I have to develop an automated code that enables me to always start from the very same initial point. By using a predefined starting point, all

further test carried out within the test bench will be easily reproducible and therefore comparable. Again, validation is "the action of checking or proving the validity or accuracy of something". I must therefore implement not a test case but a set of actions that enable me to undoubtedly affirm that my test bench is ready for testing and no further modifications or fixes shall or may be applied. If we are going to deal with physical parameters, power and performance, but also with cybersecurity matters, a main system is involved: the communication tester emulating the underlying network technology which must ensure full connectivity. To acknowledge the communications tester, or CMW500, I defined the following set of actions –here implemented as keywords in MATRIX:

- **CHECKSET GPIB STATUS**
  It searches for the comm tester (RohdeSchwarz CMW500) among all the available GPIB addresses, prioritizing the one provided by the user. If it does not work, it installs the GPIB drivers for Linux
- **CHECK COMMTESTER SOFTWARE VERSION**
  It displays the current supported versions: the 'Additional Maintenance Package', the 'BASE', and the 'Data Application'
- **DO COMMTESTER RECALL**
  It recalls the comm tester settings by loading the specified ".dfl" that must be already stored in the comm tester
- **SET COMMTESTER CONFIG**
  It loads the user configuration from the "settings.yml" file, such as IP address, type of DNS, noise, external attenuation, DL and UL powers, frequencies, and so forth
- **CHECKSET DNS**
  It tests the primary and foreign DNS server by using the configuration service app
- **CHECK INTERNET CONNECTIVITY**
  It uses the comm tester PING functionality from the Data Application to test ICMP from the comm tester towards an external Internet IP address
- **CHECKSET GO TO LOCAL**
  It sends the GPIB command 'GTL' to the comm tester to return control over the user after having used the remote control

Figure 4.5 shows how a series of keywords are automatically executed one after another. They serve as an "enabler" set, implemented as a test case for simplification purposes. Not only it checks if every requirement is met, but also installs or updates drivers if necessary. Previous set of actions was fully described here in a list basis, but I will avoid doing it again further in this document. It is highly dense and too specific. For every TC, we will need to fully design and code a set of keywords. The code related to this TC is available in Appendix D under the name "TC_SET_DEFAULT_CONFIG_CMW500.robot" –see Section D.3. I managed to successfully verify the system. If something does not work during the IVC validation process, it will not be due to the test bench. It is now time to start with the most relevant step: the **results**.

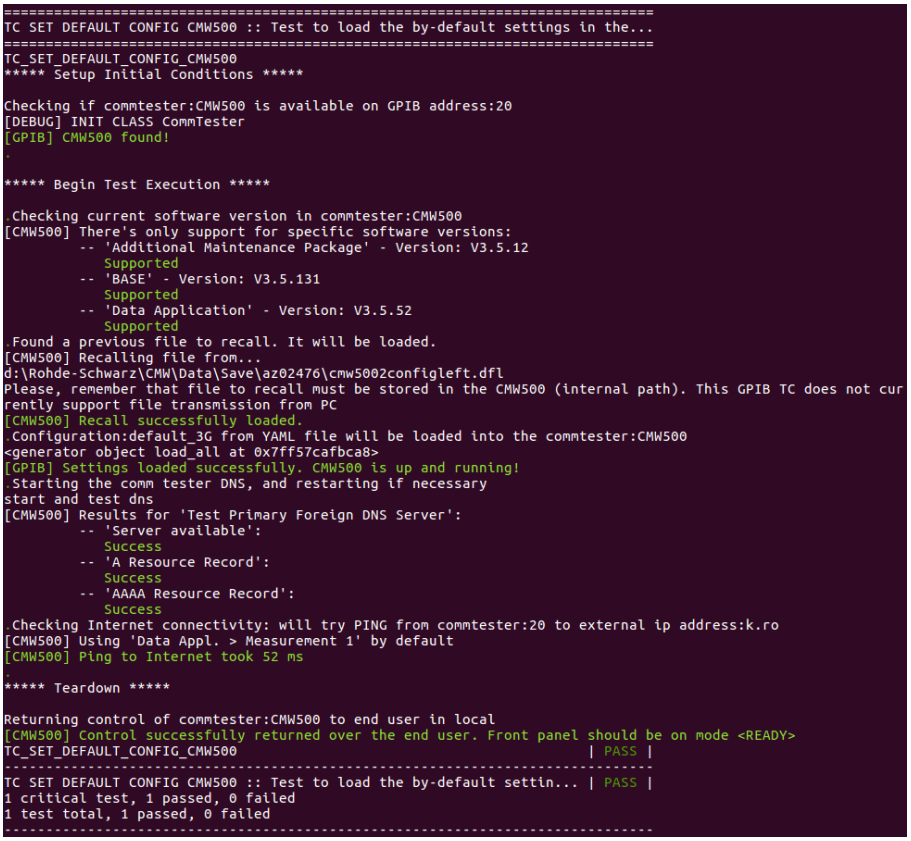


Fig. 4.5 Output of TC in MATRIX

Important!

For illustration purposes, some code related to TCs and keywords may be found on Appendix D. However, most of it is not available in order to comply with required confidentiality at RSWL

Chapter 5

Cybersecurity and performance test cases: results and discussion

The present Chapter presents all the development and automation that I achieved during this project. Basically, it is divided into two well defined parts: cybersecurity, and performance. While the former focuses on the system as a whole –the IVC and the data link between the IVC and VNEXT, the latter focuses on the IVC only, the vehicle's on-board component that brings full Internet connectivity. I wanted to study the IVC, or car modem, from the two different perspectives. Once I fully automate the physical parameters and measurement methods by remote controlling the comm tester, I will be able to build a robust system that is able to study anomaly detection, and cybersecurity attacks by gathering not only network data but also physical data –i.e. cyber attacks may be carried out when faded signalling while crossing a tunnel or while crossing an international border while in cell handover mode.

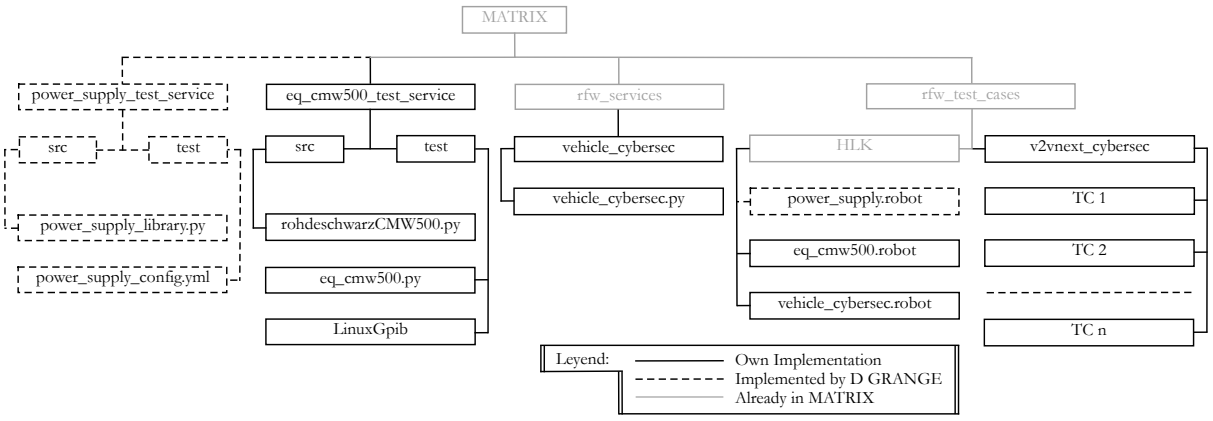


Fig. 5.1 Scheme of part of the MATRIX project

Figure 5.1 shows in a box diagram my own implementations. As I was in charge of the full automation of the test bench –presented earlier in Chapter 4, I created and developed the so called "eq_cmw500_test_service". It took me several weeks to find all required commands to remotely control the (eq)uipment and learn how to code it and send it through General Purpose Interface Bus (GPIB). Besides, we may also find the vehicle service, or "vehicle_cybersec". It has all Python methods

related to connectivity and cybersecurity test cases. Last, but not least, we find on the right part of Figure 5.1 my collaboration to the folder "rfw_test_cases". It is a common MATRIX folder where all developers include their keywords and test cases developed in the Robot framework. I used the "power_supply.robot"[1] to create new keywords allowing different consumption measurements.

I will present the foundations of my work, initially introducing the TC –or test case, requirements –or "RE", in order to present the results and analyze them later on. All analysis related to different data science aspects were carried out in "Jupyter Notebook", an "open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text" [37].

5.1 Cybersecurity

5.1.1 Test requirements and test design

Cybersecurity tests are based on different requirements, "RE" here. They mainly come from two complementary sources: internal, from RSWL –coming from previous works [8] and personal findings, and external, from the TCR. All in all, they provide a simple but powerful tool for validating some cybersecurity aspects of the IVC. As a common practice, all TC have a common **teardown**[2] where, after all automated actions have been successfully carried out, the "REMOTE" control of the comm tester is switched to manual mode, or "READY". However, **setup**[3] initial conditions differ from one TC to another. I will be centering my efforts on demonstrating that the IVC is protected when accessing the external IP network.

5.1.2 Test implementation and automation

RE 1.1: connectivity

The IVC must be able to establish a secure connection with VNEXT if the certificates are legit –meaning correct. More specifically, it will connect to MQTT, an internet broker server placed immediately before VNEXT. It handles all incoming secure connections towards the Renault's server, filtering untrusted data. For that, I must ensure the IVC has connection access to the Internet. I will initially activate the CAN bus, reboot the IVC, check that it connects via 3G to the comm tester, and finally verify it has IP connection to the Internet. This is the first and most basic TC within this project.

Figure 5.2 shows all automated keywords executed sequentially. They are all order-dependant, for the next keyword will only work if the previous one was successfully run. We say one keyword enables the later. In this TC, setup initial conditions are based on making the vehicle's modem ready to test connectivity. It will be connected to the emulated BS, the comm tester. More precisely: once the CAN

[1] Service entirely developed by Dominique.
[2] Cleanup, last step after each test method completes.
[3] To set up initial state for all test methods.

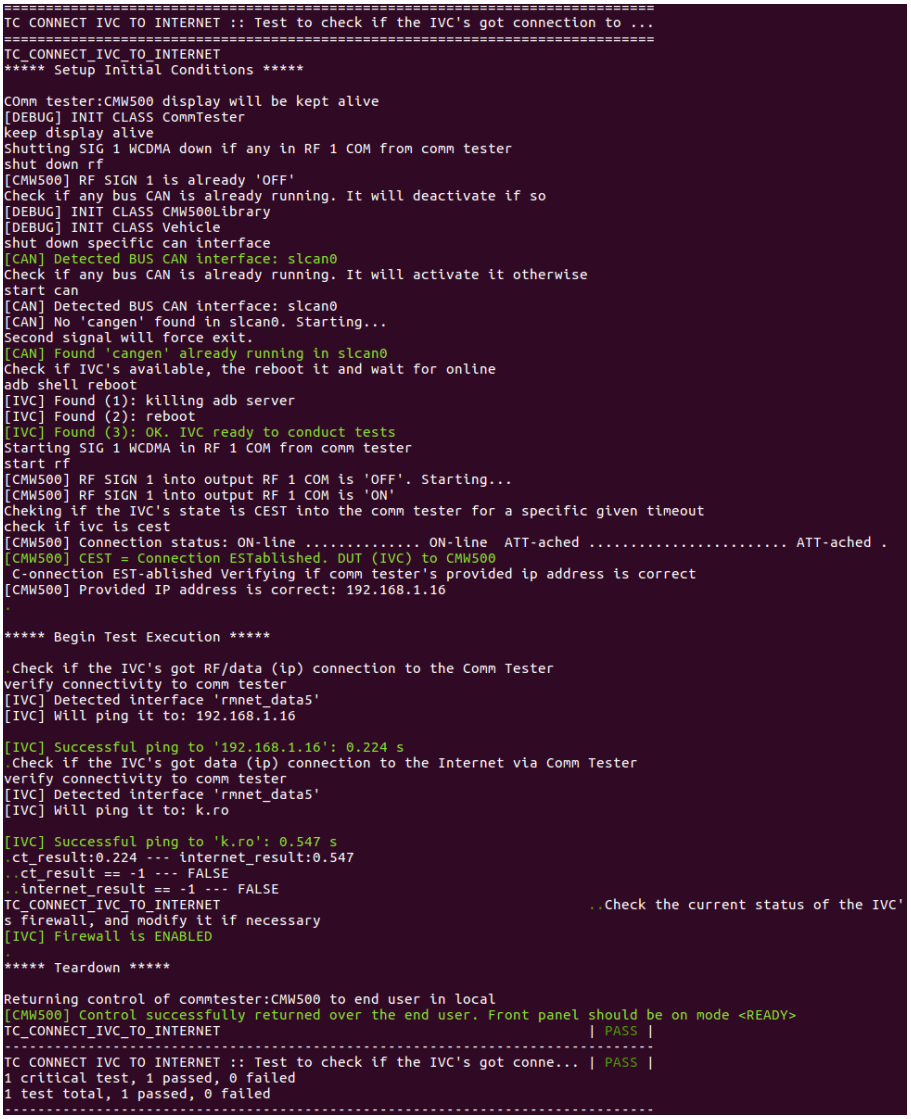
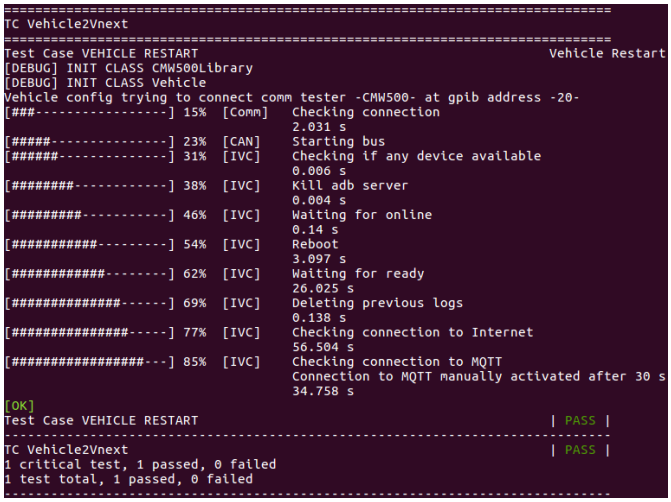


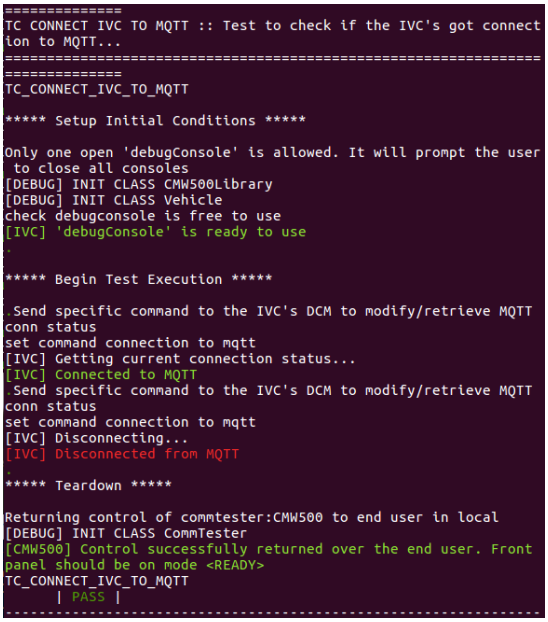
Fig. 5.2 Output of connectivity TC in MATRIX. Code is available at Appendix D

bus is activated –meaning CAN messages are being written to the bus interface “slcano”, the IVC is online and ready to carry out further network actions. By means of a GPIB interface, the keyword uses an iterative request to inquiry the comm tester via GPIB about the IVC’s connection status. These status are: offline, online, attached and connection established, or . Once the IVC is , two different PING start: (1) the IVC pings the comm tester meaning there is data connection within the RF link, and (2) pings any given Internet domain –to validate the DNS system being here the Romania-based name “k.ro”. I check here that a “PING REQUEST” is followed by a “PING REPLY”. The IVC’s firewall status is checked too. It simply checks how do the interfaces behave in front of incoming data traffic by means of Linux “iptables”. They can either “ACCEPT” or “DROP” incoming packets. By default, all IP packets coming from the RF interface “rmnet_data5” are drop if they come from untrusted sources. It is worth noting that, if enabled, the IVC’s firewall will prevent it from answering to any external “PING REQUEST”. Once everything works as intended –shown in green in Figure 5.2, the TC is PASS.

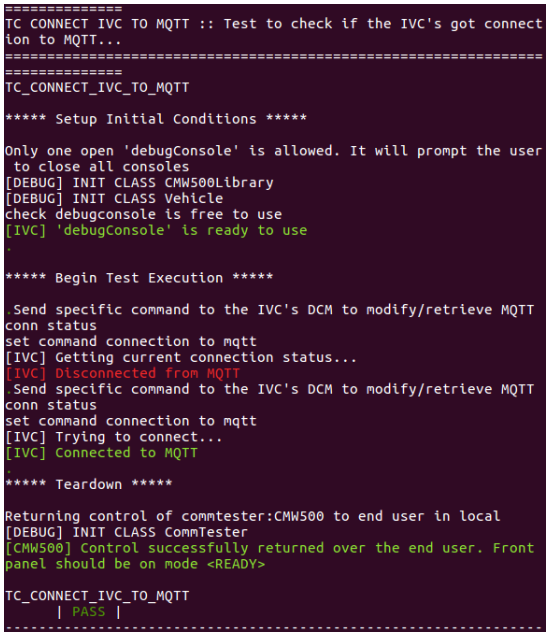
Even though I initially also checked the IVC-to-VNEXT connection status, as shown in Figure 5.3a, I decided to split it into another different TC so I could precisely control it and modify it at will.



(a)



(b)



(c)

Fig. 5.3 Output of IVC connection to MQTT TC in MATRIX

Figures 5.3b and 5.3c show how I am able to control the connection status via TC. The IVC uses its Data Connection Manager (DCM)[4] to modify its connection status at will. They are: CMAT, or "test retry strategy" meaning to try to establish a secure TLS connection to MQTT, the VNEXT broker, DM, or "disconnect MQTT broker", and GMS, or "get MQTT status", either connected or disconnected. More specifically, Figure 5.3b shows a TC that disconnects the IVC from MQTT, while Figure 5.3c shows how to connect it. Assuming all cryptography mechanisms are valid, I verify that all TLS exchanges and DCM messages are correct. They are "CONNECT" and "CONNACK", to name but a few. Besides, I use the Linux "cat" and "grep" functions to obtain information from the IVC's DCM and to filter it.

[4]It is the on-board communication device within the IVC that handles all connections with the remote server.

RE 1.2: secure TLS version

VNEXT must reject any connection coming from the IVC if the TLS version offered by the client is unsupported. As previously stated [8], minimum software versions must be required in order to assure protection against known cyber attacks. For such purpose, I use "OpenSSL", a robust, commercial-grade, and full-featured toolkit for the TLS protocol –kindly reminder: a quick TLS overview may be found on Appendix B.1. It comes with any Linux distribution, so we can find it on both PC 1 and the IVC –an IVC is just an underlying OS Linux with Android on top. Besides, we use Wireshark to capture all traffic involved.

```
================================================================================
TC CYBERSEC TLS MQTT REJECTS V10 ACCEPTS V12 :: Test to check if the MQTT b...
================================================================================
TC_CYBERSEC_TLS_MQTT_REJECTS_V10_ACCEPTS_V12
***** Setup Initial Conditions *****

Check the current status of the IVC's firewall, and modify it if necessary
[DEBUG] INIT CLASS CMW500Library
[DEBUG] INIT CLASS Vehicle
[IVC] Firewall is ENABLED
True
Only one open 'debugConsole' is allowed. It will prompt the user to close all consoles
check debugconsole is free to use
[IVC] 'debugConsole' is ready to use
.

***** Begin Test Execution *****

.Send specific command to the IVC's DCM to modify/retrieve MQTT conn status
set command connection to mqtt
[IVC] Getting current connection status...
[IVC] Disconnected from MQTT
..It will force a TLS connection with a specific protocol version towards the desired IP:port
force tls connection
[IVC] Starting TLSv1.0 handshake towards 52.233.180.235:8883
[IVC] TCP connection established!
[IVC] Not using a secure protocol: TLSv1
[IVC] No TLS errors
MQTT broker (in AVNEXT) refused the TLS connection
.Send specific command to the IVC's DCM to modify/retrieve MQTT conn status
set command connection to mqtt
[IVC] Getting current connection status...
[IVC] Disconnected from MQTT
..It will force a TLS connection with a specific protocol version towards the desired IP:port
force tls connection
[IVC] Starting TLSv1.2 handshake towards 52.233.180.235:8883
[IVC] TCP connection established!
[IVC] Using a secure protocol: TLSv1.2
[IVC] Unable to verify the first certificate
MQTT broker (in AVNEXT) accepted the TLS connection
.Send specific command to the IVC's DCM to modify/retrieve MQTT conn status
set command connection to mqtt
[IVC] Getting current connection status...
[IVC] Disconnected from MQTT
TC_CYBERSEC_TLS_MQTT_REJECTS_V10_ACCEPTS_V12                          ..
***** Teardown *****

Checking if file 'openssl_client_logs.txt' already exits in current folder, deleting it if positive. It also
 deletes the file in the IVC's 'tmp/' folder.
delete openssl files
File Exists in: /home/renault/PycharmProjects/ConnectedCar/matrix2.10/matrix
Deleting...Deleted
Returning control of commtester:CMW500 to end user in local
[DEBUG] INIT CLASS CommTester
[CMW500] Control successfully returned over the end user. Front panel should be on mode <READY>
TC_CYBERSEC_TLS_MQTT_REJECTS_V10_ACCEPTS_V12                          | PASS |
--------------------------------------------------------------------------------
TC CYBERSEC TLS MQTT REJECTS V10 ACCEPTS V12 :: Test to check if t... | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
--------------------------------------------------------------------------------
```

Fig. 5.4 Output of Secure TLS version TC in MATRIX. Code is available at Appendix D

Figure 5.4 presents the automated TC related to RE 1.2. Using previous keywords from RE 1.1, it will initially disconnect the IVC from MQTT if already connected. It will then try to establish two different TLS sessions one after the other. The IVC will try to connect VNEXT via TLS 1.0 and TLS 1.2 versions.

As shown in red from Figures 5.4 and 5.5a, the "MQTT broker refused the TLS connection" while in version 1.0 –see a MQTT's "TLS FIN" right after the IVC's "Client Hello" message. On the contrary, the "MQTT broker accepted the TLS connection" while in version 1.2 –see Figure 5.5b. Unlike the previous case, there are no end of TLS connection messages. Therefore, the TC is PASS given that MQTT behaves securely with regard to TLS, rejecting non-recommended *v*1.0 but accepting *v*1.2.

192.168.1.47	52.233.180.235	TCP	74 39437 → 8883 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=257237 TSecr=0 WS…
52.233.180.235	192.168.1.47	TCP	74 8883 → 39437 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1440 WS=256 SACK_PERM=1 TSval=…
Netgear_a8:4b:ee	00:e1:00:00:16:6a	ARP	60 192.168.1.1 is at a0:40:a0:a8:4b:ee
192.168.1.47	52.233.180.235	TCP	66 39437 → 8883 [ACK] Seq=1 Ack=1 Win=29216 Len=0 TSval=257251 TSecr=803283253
192.168.1.47	52.233.180.235	TLSv1	168 Client Hello
52.233.180.235	192.168.1.47	TCP	66 8883 → 39437 [FIN, ACK] Seq=1 Ack=103 Win=263424 Len=0 TSval=803283404 TSecr=257253
192.168.1.22	192.168.1.255	NBNS	92 Name query NB ISATAP<00>
192.168.1.47	52.233.180.235	TCP	66 39437 → 8883 [ACK] Seq=103 Ack=2 Win=29216 Len=0 TSval=257267 TSecr=803283404
192.168.1.47	52.233.180.235	TCP	66 39437 → 8883 [FIN, ACK] Seq=103 Ack=2 Win=29216 Len=0 TSval=257267 TSecr=803283404
52.233.180.235	192.168.1.47	TCP	66 8883 → 39437 [ACK] Seq=2 Ack=104 Win=263424 Len=0 TSval=803283544 TSecr=257267

(a)

192.168.1.47	52.233.180.235	TCP	74 39438 → 8883 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=258275 TSecr=0 WS…
52.233.180.235	192.168.1.47	TCP	74 8883 → 39438 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1440 WS=256 SACK_PERM=1 TSval=…
192.168.1.47	52.233.180.235	TCP	66 39438 → 8883 [ACK] Seq=1 Ack=1 Win=29216 Len=0 TSval=258288 TSecr=1514848333
192.168.1.47	52.233.180.235	TLSv1.2	242 Client Hello
52.233.180.235	192.168.1.47	TLSv1.2	1389 Server Hello, Certificate, Server Key Exchange, Certificate Request, Server Hello Done
192.168.1.47	52.233.180.235	TCP	66 39438 → 8883 [ACK] Seq=177 Ack=1324 Win=32096 Len=0 TSval=258307 TSecr=1514848476
192.168.1.47	52.233.180.235	TLSv1.2	171 Certificate, Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
52.233.180.235	192.168.1.47	TLSv1.2	117 Change Cipher Spec, Encrypted Handshake Message
192.168.1.47	52.233.180.235	TCP	66 39438 → 8883 [ACK] Seq=282 Ack=1375 Win=32096 Len=0 TSval=258334 TSecr=1514848774

(b)

Fig. 5.5 Wireshark traffic captures while different TLS sessions: (a) TLS v1.0, (b) TLS v1.2

RE 1.3: Denial-of-Service

The IVC must be resilient to external denial-of-service attacks that are based on data flooding. Its external connectivity must be assured while its debugging console must remain available under any circumstance. For that, counter measurements must be set. In theory, the IVC's enabled firewall should wholly protect it.

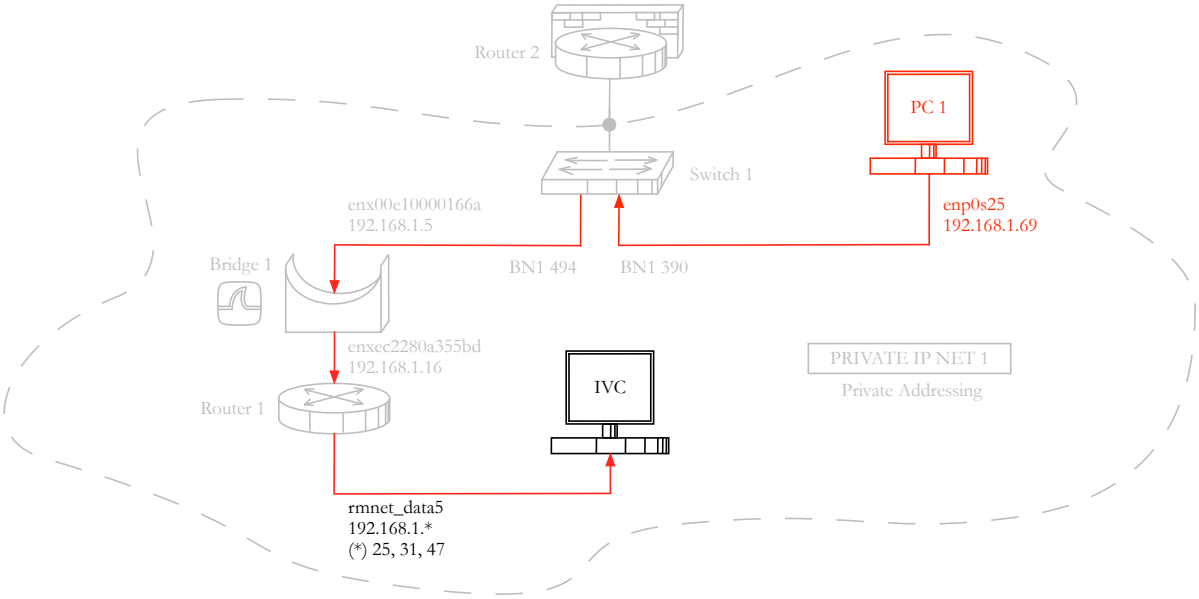


Fig. 5.6 DoS attack network layout

Globally known, a Denial-of-Service (DoS) attack sole purpose is to overwhelm a specific server and the firewall protecting it. Its principle is simple: to send a large number of data targeting a server so that the device's ability to process incoming data and respond get blocked. We say that the device, service or system under attack "freezes" –there is no external connectivity. If the attack comes from multiple, synchronized, distributed sources, we then say it is Distributed Denial-of-Service (DDoS). Such attacks are normally based on User Datagram Protocol (UDP) [38, 39], a very well-known ISO layer 4 connectionless protocol for establishing low-latency and loss-tolerating connections. Running

on top of IP, consequently also referred to as UDP/IP, it provides process-to-process communications via messages, or datagrams, requiring a port number. It is ideal for voice, video and gaming applications. Alternatively we find Transmission Control Protocol (TCP) [40, 39], another very well-known ISO layers 4 and 5 connection-oriented protocol for enabling flow control and establishing error-free data transmissions. Running on top of IP, consequently referred to as TCP/IP, it provides host-to-host communications. It is ideal for messaging and secure exchanges.

For this RE 1.4 and as shown in Figure 5.6, we will basically require (1) a functional, online **IVC with enabled firewall**, in black, and (2) a regular **PC with connectivity access**, in red, located in the same intranet than the previous IVC. It is PC 1 the one that will carry out the DoS attack –its simple but powerful Python code may be consulted on Appendix D.

Fig. 5.7 Output of DoS attack TC in MATRIX

Figure 5.7 shows how I implemented the DoS attack. Right after the IVC's connectivity is verified, and its "adb shell" is alive, PC 1 starts the DoS UDP flooding attack. It results in a blocked IVC whose "adb shell" is dead and has no external connectivity. I use a multiprocess code for testing it up. It remains blocked as long as the attack takes place. No PING messages, TCP connections, or any other data exchange are possible. Therefore, the result of this particular TC is clearly FAIL. For a given software version –IVC's current running version is 12.1, the IVC is not protected against flooding attacks. Some direct consequences would be to make the IVC unable to transmit any warning message in case of danger, or any position or environment information in case of dense traffic while in assisted driving. It would also mean no FOTA and no remote services. Thanks to the joint capabilities of "Scapy", a packet manipulation tool for computer networks [41], and Wireshark, the packet capturing tool, I can analyze all incoming traffic. It allows me to apply some data science techniques in order to characterize such traffic and try to ignore it if necessary –further readings related to DoS attack detection based on ML algorithms may be found in [42].

For example, let me introduce you the Figure 5.8, containing two different Principal Component Analysis (PCA) [43] analysis applied over the incoming traffic to the IVC. Thanks to the use of PCA from "sklearn", a Python library for data science, I am able to convert correlated variables related to the traffic into uncorrelated components of smaller dimension –the so called "scaling". In this case, I take 20 different variables such as the packet id, flag, checksum, payload, protocol, to name but a few. PCA is a tool in exploratory data analysis, also ideal for making predictive models. On Figure (pca).a, we see all traffic involved during a typical connection from the IVC to MQTT as in RE 1.1. It is based on two principal components whose variation values are 36,5% for "pca-one", or first and most relevant PCA component, and 29,7% for "pca-two", or second most important PCA component. Similarly on Figure (pca).b, We show the captures right after the UDP flooding attack, with 35,5% and

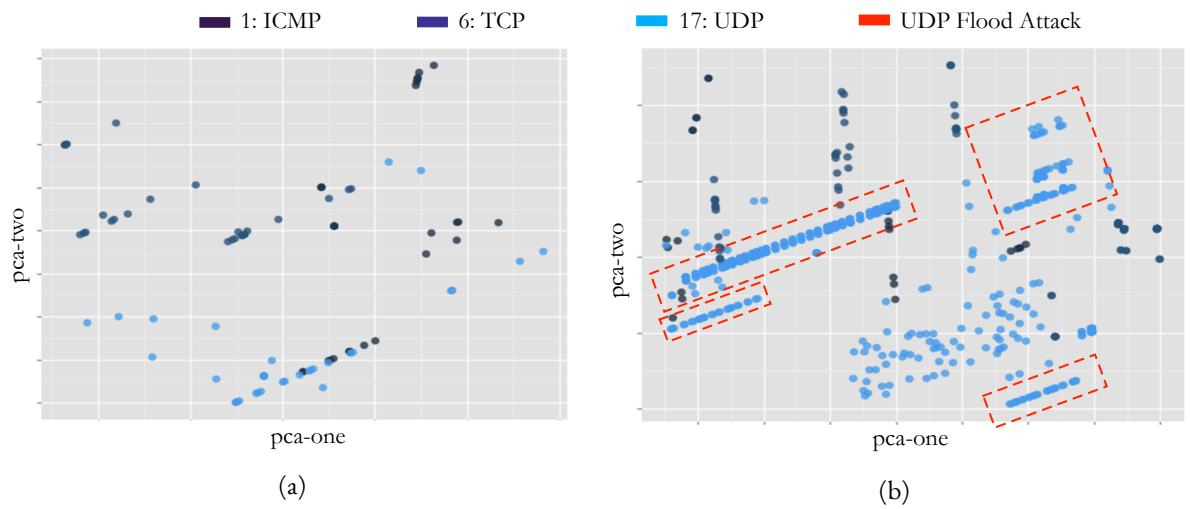


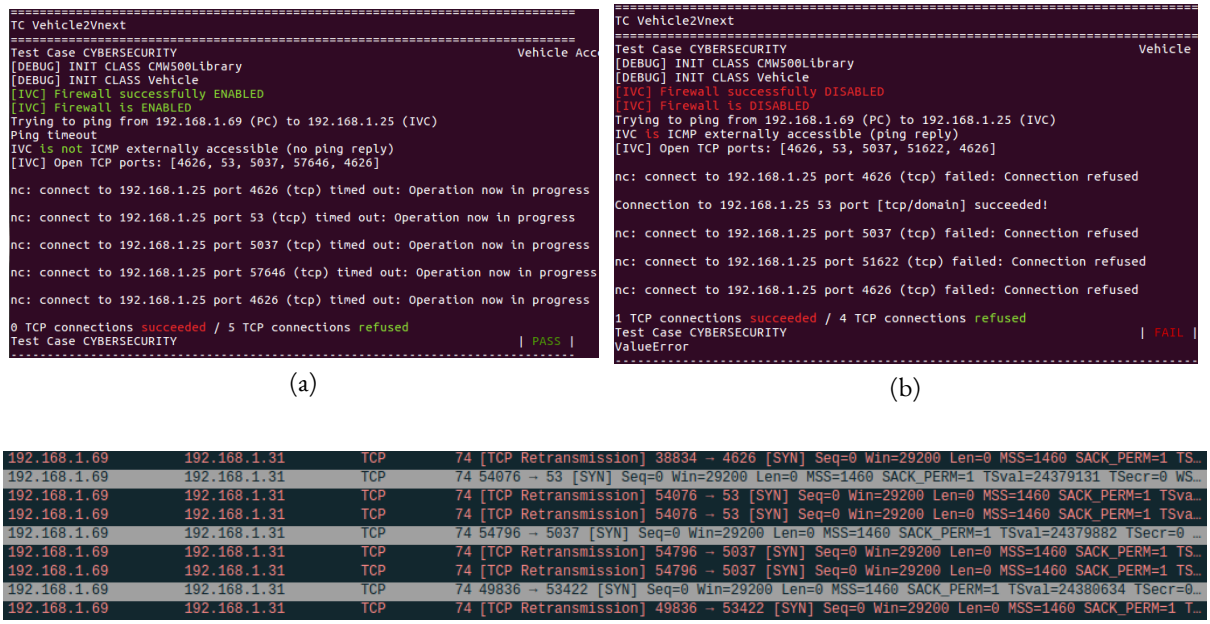
Fig. 5.8 Visualization of a DoS attack through PCA: (a) before, (b) after

27,4% respectively. One could think about creating and integrating an uncomplicated algorithm that would serve as a protection against it. Once the PCA is calculated over enough training data, it could be easily applicable to all incoming traffic as a mathematical operation. We just need to compute the mathematical coefficients and apply to test data. For example, I could use the Figure (pca).b to find patterns associated to DoS attacks. For example, we could say that they are based on dense, linearly distributed attacks whose slope into the PCA-one vs PCA-two domain is close to 0,5.

RE 1.4: accessibility

The IVC must not be accessible from the Internet –generally speaking, vehicles cannot be accessed from the Internet. For that, counter measurements must be set. Just as it happened on RE 1.3, a firewall is expected to work. When we say "accessible", we mean ICMP and TCP protocols. ICMP is an Internet Layer protocol mostly use to report errors and verify that a host may or may not be reachable. TCP is a Transport Layer protocol for connection oriented data exchanges. Therefore, I will test whether the IVC ignores both protocols or not. I will use "adb shell" to request for the open TCP ports. Even though they may be open, the firewall should DROP any incoming TCP SYN trying to establish a TCP connection –for example someone trying to establish a secure TLS connection. I use "netcat", a Linux computer networking utility for reading from and writing to network connections using TCP, for establishing such connections.

Figure 5.9 shows the result of this particular TC. When the IVC's firewall is enabled –as in Figure 5.9a, no PING –"timeout", or TCP SYN messages –"(tcp) timed out", are answered. If either the PING nor the TCP protocol succeed, the TC is PASS. I then affirm the IVC's accessibility is well protected. We can see in Figure 5.9c how the external "attacker" –the one who is trying to connect to the car's modem, tries to retransmit all unanswered TCP SYN messages from 192.168.1.69. In this case, the IVC has two different IP addresses, 192.168.1.25 in 5.9a and 5.9b, and 192.168.1.31 in 5.9c, due to DNS dynamic address assignment –screenshots were taken on two different days. On the contrary, I present to you the Figure 5.9b whose TC is FAIL. After the firewall has been disabled, both PING –"ping reply", and TCP SYN messages –"Connection (...) succeeded", get a response. However, it does not



(a) (b)

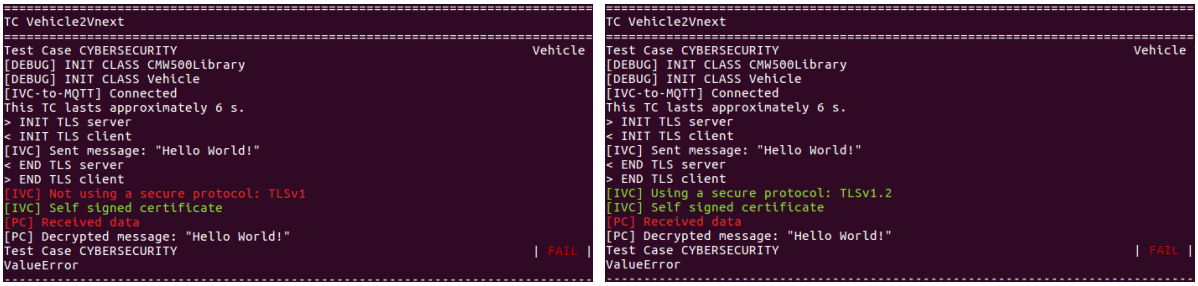
(c)

Fig. 5.9 Output of IVC accesibility TC in MATRIX

imply the IVC is unsecure. Once a firewall is disabled, no protection measurements are present at the system. Obtaining a TC FAIL when the firewall is disabled is reasonable, yet unpleasant.

RE 1.5: secure accessibility

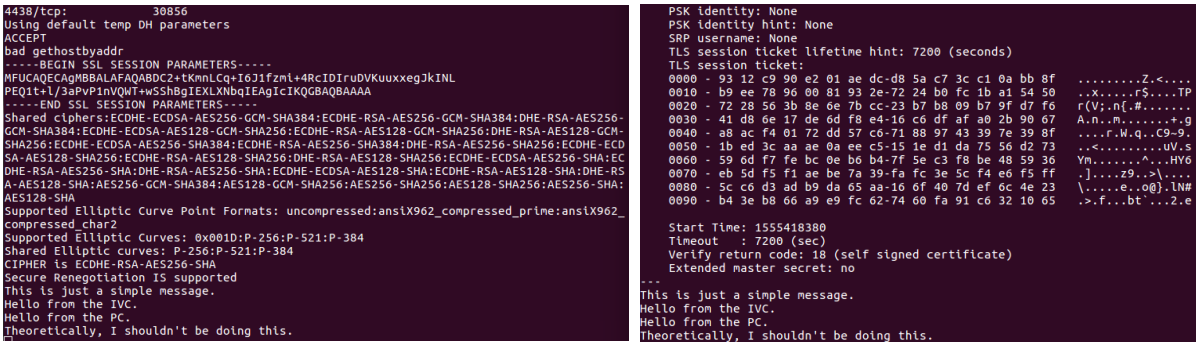
The IVC, as client, must reject any TLS secure connection if the server's certificates are unknown, self-signed, expired, or corrupted. Given that I obviously cannot control either MQTT ot VNEXT at will, I will be using an alternative TLS server for dealing with secure connections. Basically, I will use OpenSSL to define a listening TLS server on PC 1 at port 4438. For that, I will use the "s_server" option and all required keys and certificates. I will be using our own server so we need to define a self-signed certificate. I will force the IVC to connect to the fake server by means of "s_client".



(a) (b)

Fig. 5.10 Output of TLS connection to a fake server TC in MATRIX: (a) server forces TLS v1.0, (b) server forces TLS v1.2

Figure 5.10 shows how the IVC, when forced to, establishes a TLS connection with a foreign server even though its certificates are self-signed. As shown in Figure 5.10a, the IVC accepts insecure TLS versions, such as the not-recommended-to-use v1.0. I am also able to write data into the IVC's console –see decrypted message.



(a) (b)

192.168.1.31	192.168.1.69	TCP	74	53421 → 4438 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=8924181 TSecr=0 W…
192.168.1.69	192.168.1.31	TCP	74	4438 → 53421 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=236593…
192.168.1.31	192.168.1.69	TCP	66	53421 → 4438 [ACK] Seq=1 Ack=1 Win=29216 Len=0 TSval=8924193 TSecr=23659395
192.168.1.31	192.168.1.69	TLSv1	242	Client Hello
192.168.1.69	192.168.1.31	TCP	66	4438 → 53421 [ACK] Seq=1 Ack=177 Win=30080 Len=0 TSval=23659430 TSecr=8924195
192.168.1.69	192.168.1.31	TLSv1	2239	Server Hello, Certificate, Server Key Exchange, Server Hello Done
192.168.1.31	192.168.1.69	TCP	66	53421 → 4438 [ACK] Seq=177 Ack=1449 Win=32096 Len=0 TSval=8924212 TSecr=23659432
192.168.1.31	192.168.1.69	TCP	66	53421 → 4438 [ACK] Seq=177 Ack=2174 Win=35008 Len=0 TSval=8924214 TSecr=23659432
192.168.1.31	192.168.1.69	TLSv1	200	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
192.168.1.69	192.168.1.31	TLSv1	300	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
192.168.1.31	192.168.1.69	TCP	66	53421 → 4438 [ACK] Seq=311 Ack=2408 Win=37888 Len=0 TSval=8924233 TSecr=23659498
192.168.1.31	192.168.1.69	TLSv1	156	Application Data, Application Data
192.168.1.31	192.168.1.69	TLSv1	103	Encrypted Alert
192.168.1.69	192.168.1.31	TCP	66	4438 → 53421 [ACK] Seq=2408 Ack=439 Win=31104 Len=0 TSval=23659530 TSecr=8924235
192.168.1.69	192.168.1.31	TCP	66	4438 → 53421 [FIN, ACK] Seq=2408 Ack=439 Win=31104 Len=0 TSval=23659530 TSecr=8924235
192.168.1.31	192.168.1.69	TCP	66	53421 → 4438 [ACK] Seq=439 Ack=2409 Win=37888 Len=0 TSval=8924247 TSecr=23659530

(c)

Fig. 5.11 Output of IVC accesibility TC in MATRIX

Precisely, Figure 5.11 shows how a TLS connection can be established between the IVC –192.168.1.31 on Figure 5.11b, and the fake TLS server running on PC 1 –192.168.1.69 on Figure 5.11a. At the bottom of both images we find a simple toy dialogue[5] where the "Application Data" occur –as in Figure 5.11c representing the captured traffic.

5.1.3 Results and discussion

Previous sections allowed a simple but powerful analysis with regard to the cybersecurity of the IVC. For a given software version, I demonstrated that the IVC:

- Cannot be externally accessible if firewall enabled
- Gets blocked after a DoS attack even if firewall enabled
- Can establish a TLS session with untrusted server and non-recommended protocol version

Besides, I checked that VNEXT –via MQTT, does not accept incoming TLS sessions when asking for non-recommended protocol versions such as v1.0. I reported these results to the Groupe Renault cybersecurity experts at the TCR, Paris. For that, I arranged an online meeting on the 17/04/2019 with a Security and Innovation Project Manager, a Cybersecurity Engineer, and a Cybersecurity Specialist. During this RSWL/TCR meeting and thanks to the previously presented findings, we agreed to the following terms:

[5] «-This is just a simple message, +Hello from the IVC, -Hello from the PC, +Theoretically I shouldn't be doing this.»

1. I, as RSWL employee included within the MATRIX project, was able to objectively demonstrate that cybersecurity test cases can be automated and thus integrated into the test catalog of automated test cases.
2. There was a positive compromise between RSWL and TCR towards defining a common framework for collaboration to have an automated test bench for cybersecurity. A roadmap was agreed by the TCR to iteratively provide test requirements and increase their participation within the MATRIX project. Plus, an interactive on-site meeting is planned in Toulouse.
3. The TCR is interested on cybersecurity tests log reviewing.

5.2 Performance

When I say "performance" I refer to power and performance analysis. They are both covered by the Power and Performance (PnP) team who analises, optimises, debugs, and develops all test related to it. During this project, I worked as a developer in order to deliver a working product –several automated test cases, to the team, playing the role of client.

5.2.1 Test requirements and test design

Differently than previously presented, the requirements and specifications related to performance had not been yet written not released during the realization of this part of the project. Instead, the already achieved automation of the communication tester opened the possibility to carry out reliability and robustness testing on the one hand, and power and consumption testing on the other. Basically, the idea was to fully automate three different testing requirements. Instead of focusing on my TC implementations as I did before, I will visualize the TC outcomes to provide with a full tangible findings.

5.2.2 Test implementation and automation

RE 2.1: Time-To-Connection

Time-To-Connection (TTC) is the time taken by the IVC to recover the Internet connectivity once the RF link is lost for a certain period of time –therefore there is a loss of data connection. As we look for stable and consistent results, many iterations will be carried out –reliability test. In order to measure the time the IVC takes until Internet reconnection once the RF signal is off for a certain time we need to define a simple algorithm. I initially developed it into a single keyword composed by all required steps, and run into a simple TC. These are:

1. Set 3G cell to "off"
2. Start for loop:
   (a) Set 3G cell to "on"
   (b) Wait until 3G cell is verified to be "on"
   (c) Start timer
   (d) Start while loop until ping is working or ping connection timeout is up:
      i. Try ping to an Internet address
      ii. If ping "ok":
         A. Measure time through timer and write it to file
         B. Exit while loop
      iii. If ping "ko":
         A. Sleep a certain time
         B. Add taken time to timer, and keep trying 2.d.i

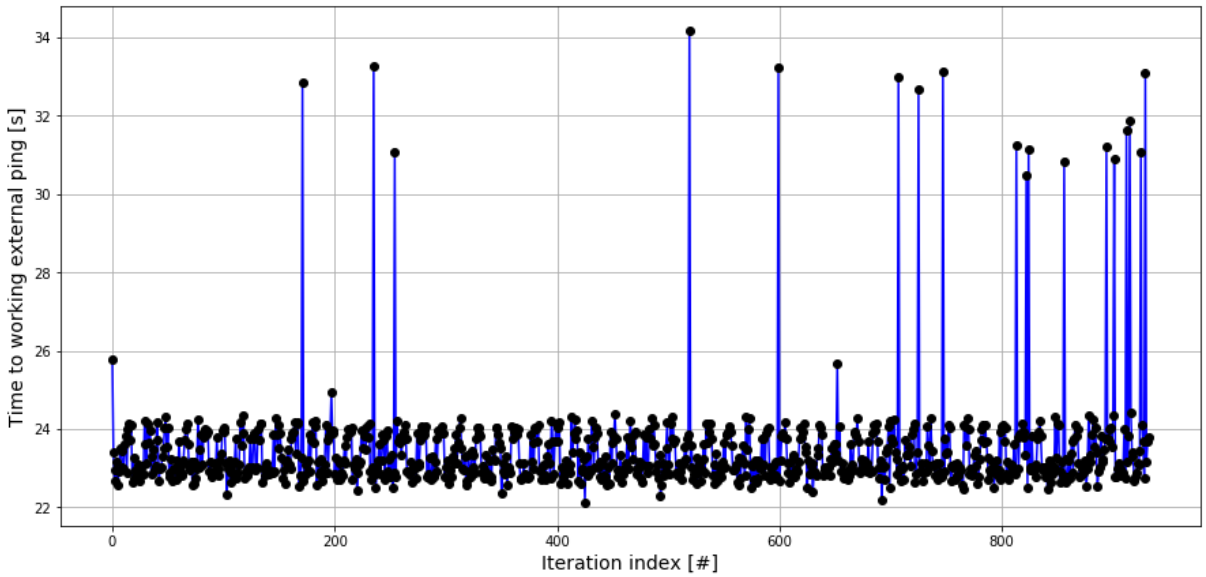


Fig. 5.12 Visualization of output of TTC TC of over 900 iterations in MATRIX

The Figure 5.12 represents the initial solution for RE 2.1 for over 900 iterations taking almost 7 hours to complete. As shown, two different behaviors are present: a generalized of around 23 seconds, and a secondary, less frequent of 31 seconds or higher. The latter tends to happen more frequently in the second half of the test. Therefore, the hourtime the test is launched may affect the results. No measurement takes longer than 35 seconds; it is an important result that will be analyzed later on. I also detect an oscillatory behavior, so I will focus on it. It is worth noting I am using a resolution of 0.1 seconds.

Figure 5.13 is a zoomed version of Figure 5.12, centered at iteration index number 400. All waves have a similar pattern, as shown in Figure 5.14. It may suggest that the IVC presents somehow a wavy pattern when trying to reconnect to the Internet, based on all mechanisms and actions needed to recover connection. The max-to-min ratio within this oscillatory pattern may be of only 1 second, but thanks to the analysis of data I found something interesting. Even though I am not able to say why this

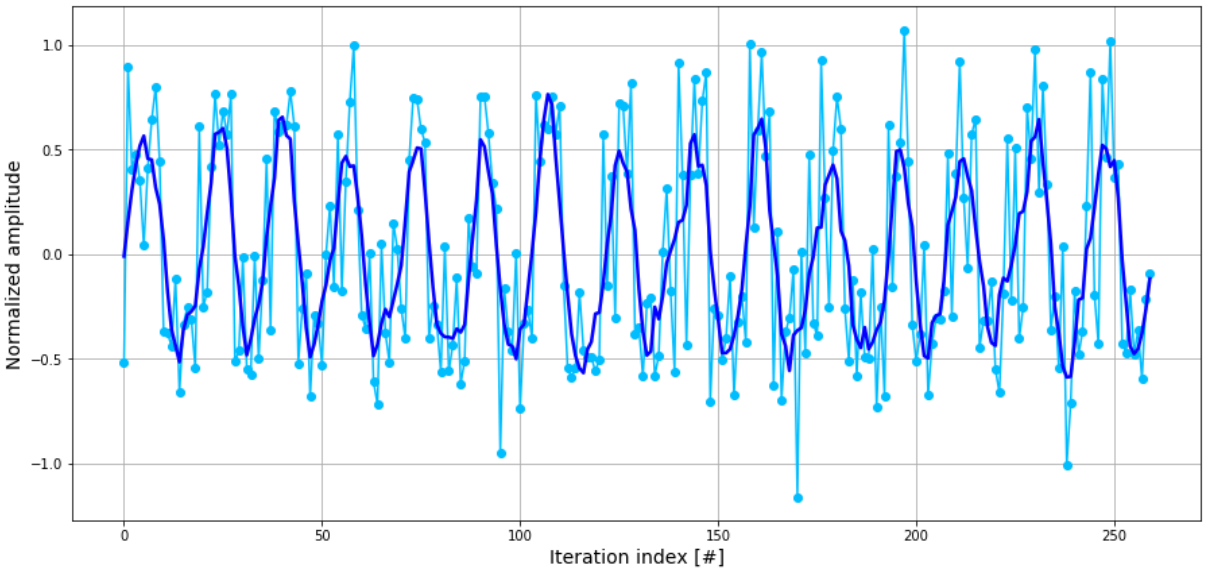


Fig. 5.13 Zoom of over 250 iterations from Figure 5.12

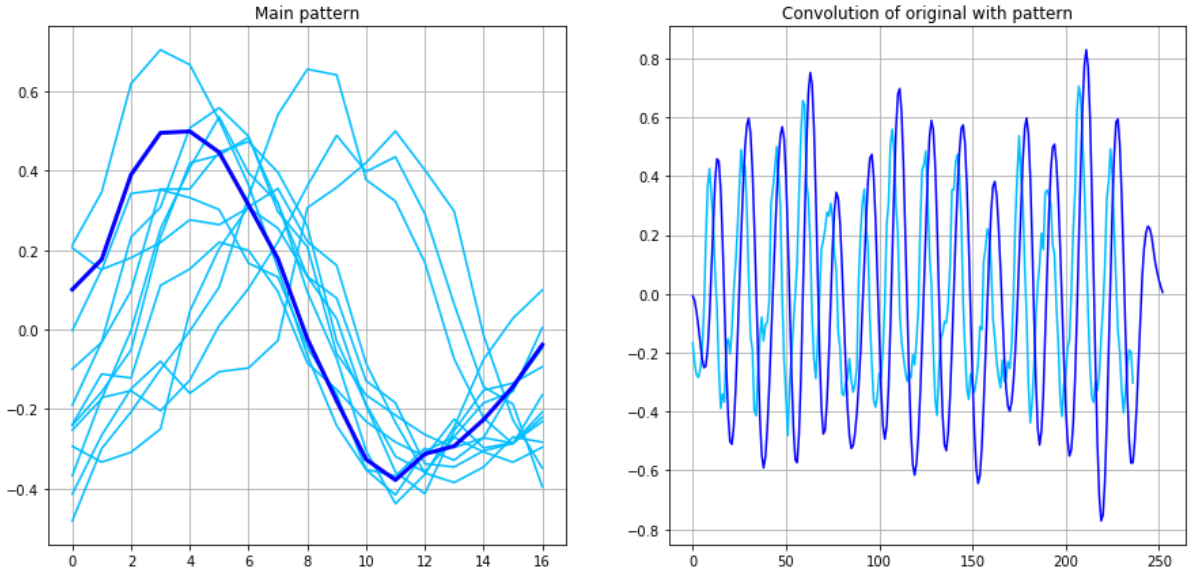


Fig. 5.14 Further analysis on iterations

happens, I can definately inform about its existence to the PnP team. I ran over 10 same tests and all had the same underlying pattern. Thus, from now own, I will decrease the TTC resolution so only steps of 1 second can be considered. The wavy pattern will then disappear.

We can consider another type of study: the prediction of time peaks. As shown in Figure 5.15 in red, initial Figure 5.12 was normalized so maximum TTC times of 30 seconds or higher are considered to be 1. Such peaks may be considered to follow a random distribution. However, I decided to implement an algorithm based on ML techniques in order to try, at least, to estimate the presence of such peaks. The idea is simple: use the samples I have to train a model on how many iterations remain until a new peak of 30 seconds or above arrives.

Figure 5.16 shows this idea. After every peak, an iteration counter is set backwards so I precisely know how many remaining iterations are left until a new peak. For example, taking the image above, there is a peak at the iteration index 1225. Consequently, a backwards iteration counter starts at 1225 whose

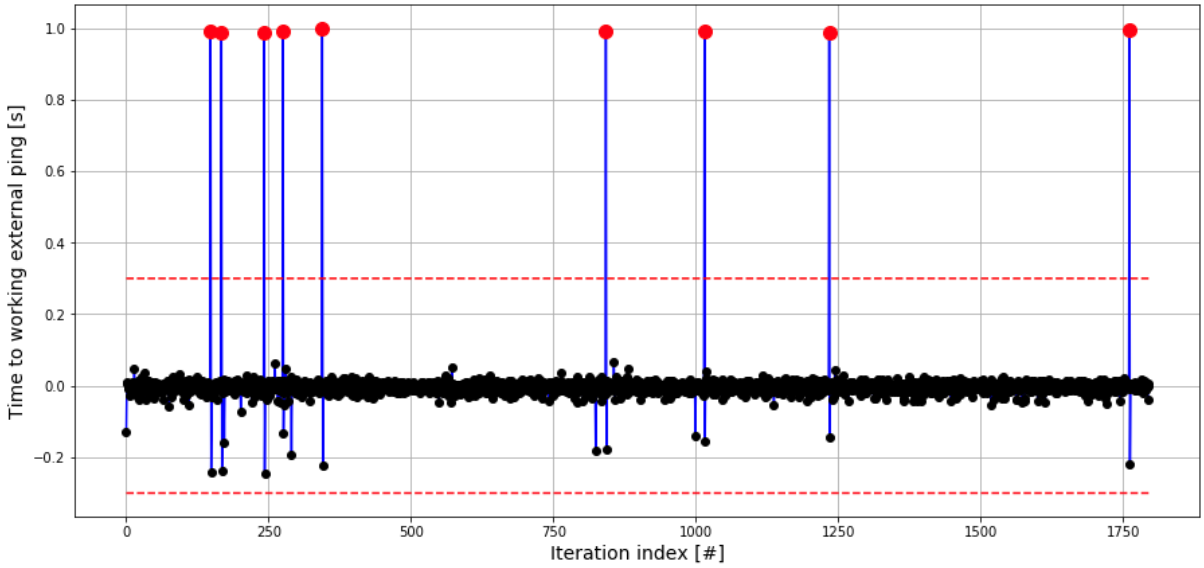


Fig. 5.15 Version of Figure 5.12 where higher peaks have been normalized to one

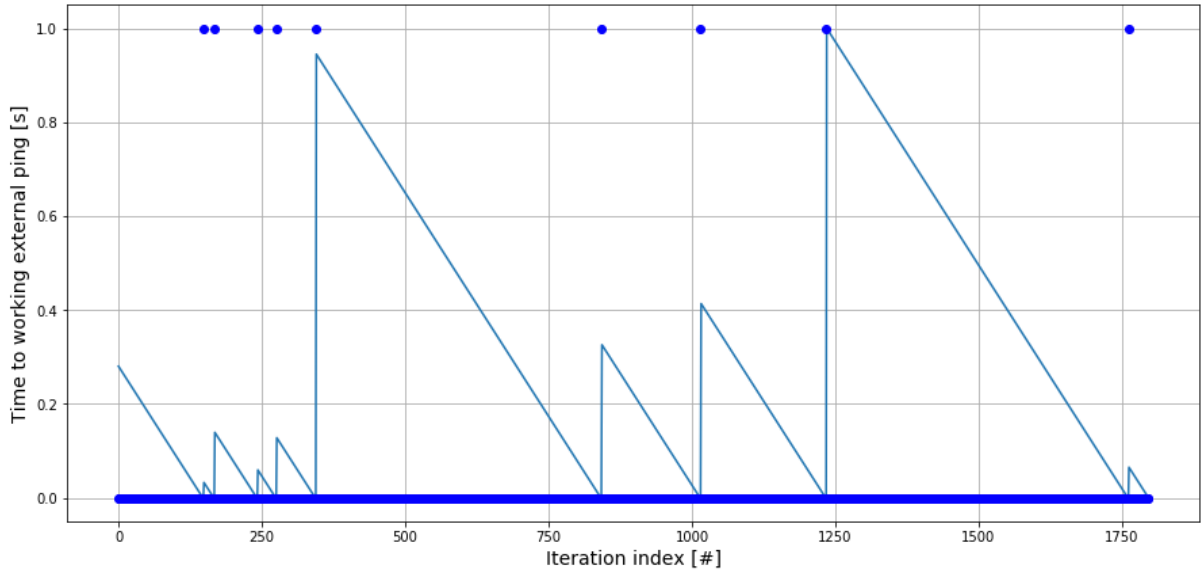


Fig. 5.16 Triangles representing time to next peak arrival

value is around 525. It will decrease 1 point at every iteration until the arrival of a next peak at iteration index 1750. This backwards iteration counter will be then 0. I use this triangular pattern, or remaining time until next peak, for data prediction.

Figure 5.17 represents a working example of this idea. On it, several ML models such as random forests or neural networks are fed with different available information related to the data I want to predict –therefore colors and labels are not relevant here. We are dealing here with a predictive maintenance project related to aircrafts –carried out as a scholar project at ISAE-SUPAERO in cooperative working teams of 3 students while in a data science Hackathon[6]. We are estimating the Remaining Useful Life (RUL) [44] value, or time in cycles before the engine fails and equivalent to our next ping peak prediction, by using data related to many different sensors located within the engine –equivalent to

[6]Design, sprint-like event in which computer programmers and others involved in software development collaborate intensively on software projects

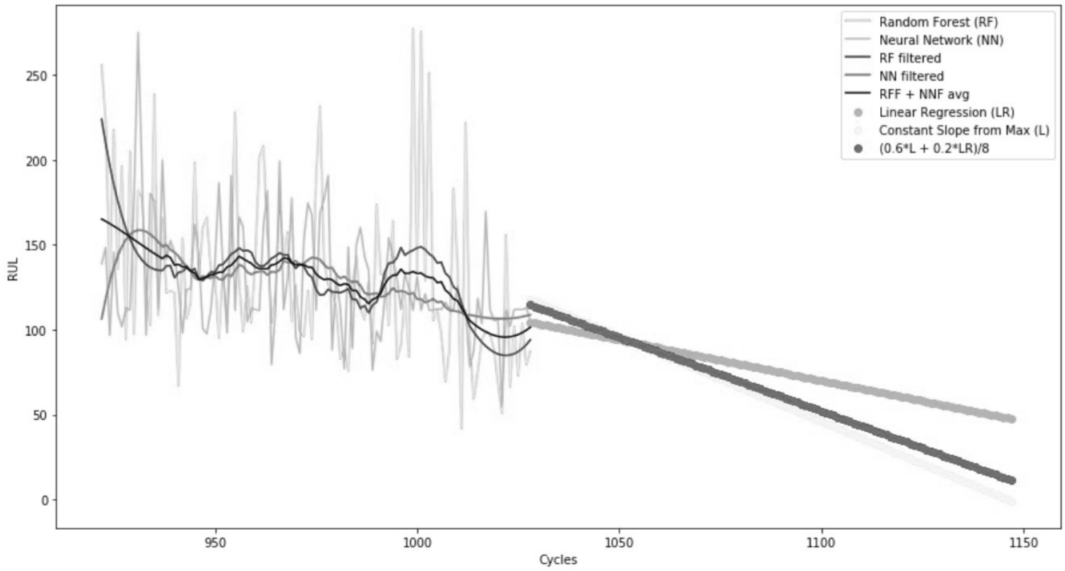


Fig. 5.17 Several coordinated ML models to predict time-series data

our supplementary information related to ping time, duration, target address, and so forth. As a result, we devise a decent slope, in dark green. We will use this slope to draw a line representing precisely the triangles we would like to predict from Figure 5.16.

Back to the topic, I used a single keyword to implement this TC. However, it does not follow the main idea of the MATRIX paradigm: building automated test based on several keywords enabling modular testing. In plain English, we have to code a robust test case where each step belongs to a different, well defined function –or keyword. Now, each action within the function must be coded into a single, **different keyword**. All the complexity is extracted from a single keyword to many. The idea behind the algorithm is the same though. This more functional, modular **programming provides multiple combinations enabling easier test case design and implementation**.

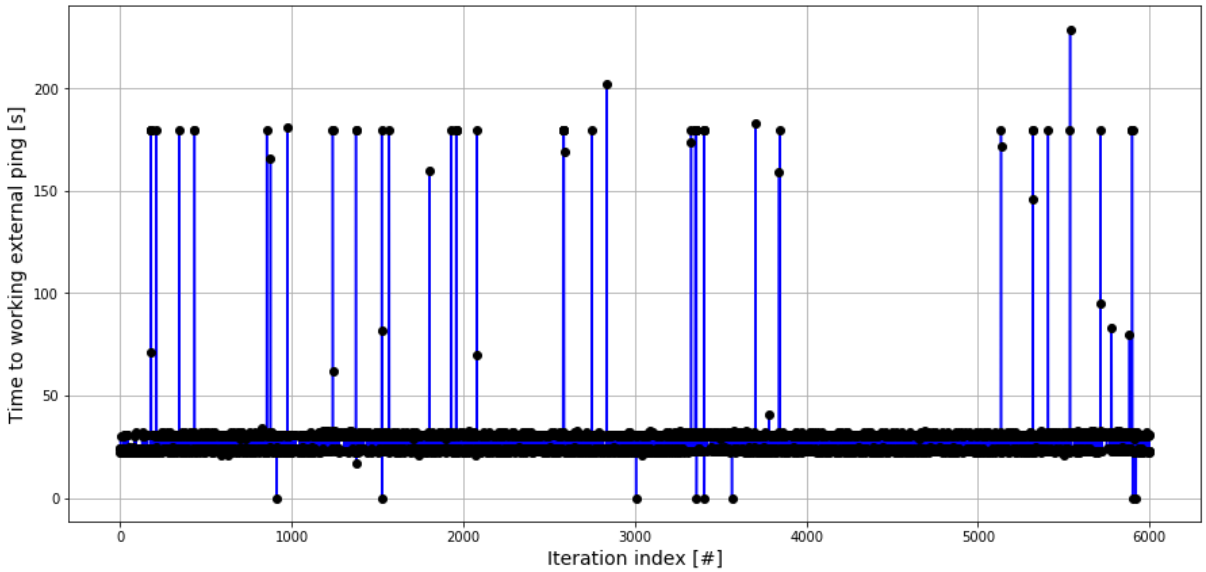


Fig. 5.18 Visualization of output of modular-through-keyword TTC TC of 6000 iterations in MATRIX

Figure 5.18 shows the first results obtained with this new TC version built on modular keyword programming and error handling. It may be seen that both 24 and 35 second behaviors are present. However, there are much more higher peaks of around 180 seconds –180 is the chosen timeout before the target address is considered unreachable. In order to obtain who was causing these unprecedented errors, I completed the TC with an error log so I could catch the different errors –full robot code related to this TC may be found on Appendix D under the name "TC_IVC_RELIABILITY_INTERNET.robot". We reduced it to two different hypothesis:

1. I am causing the GPIB to fail, but I do not know why. It may be due to using a wrong settling time, or that I am sending the wrong instructions or even right instructions but in the wrong order. Alternatively, I may be causing the GPIB interface to be overcharged. All these problems lead to the same solution: it requires a GPIB drivers full reinstallation.

2. The IVC status is not CEST, or “Connection ESTablished”, meaning it was not able to use the 3G technology over the RF interface to connect to the comm tester.

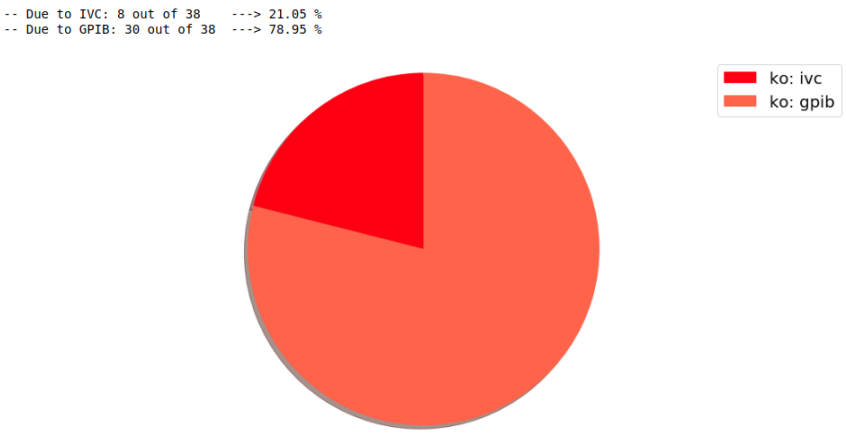


Fig. 5.19 Number of errors due to IVC and to GPIB

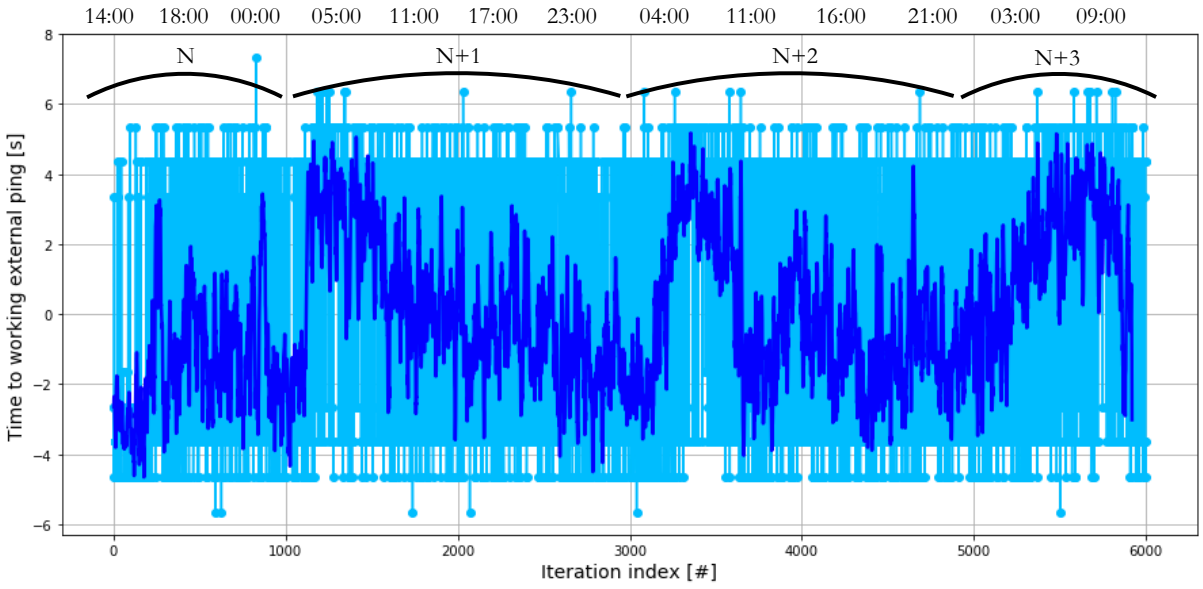


Fig. 5.20 Sliding average window over 6000 iterations

Even though 5962 out of 6000 iterations in TC from Figure 5.19 are positive and result into no errors, 38 are entirely based on either the GPIB connection or, a priori, a malfunctioning into the IVC network capabilities. It means my developed code and therefore any resulting TC is robust face errors and it may catch unprecedented bugs.

Last, but not least, I introduce the Figure 5.20. It accounts for all 5932 previous successfully iterations where their average has been subtracted, and a sliding average is shown in dark blue. It reinforces the previously presented hypothesis: "the hourtime the test is launched may affect the results". On the upper part of the image I find the hourtime, as well as the nth day as stated by the capital letter "N". As shown, the moving average tells me that the network may be densely used on the late night or early morning[7] resulting in higher TTC values.

RE 2.2: power

The IVC must set a specific output power value at his transmitting uplink RF antenna, providing stable, precise readings. In plain English, we must get what we ask for.

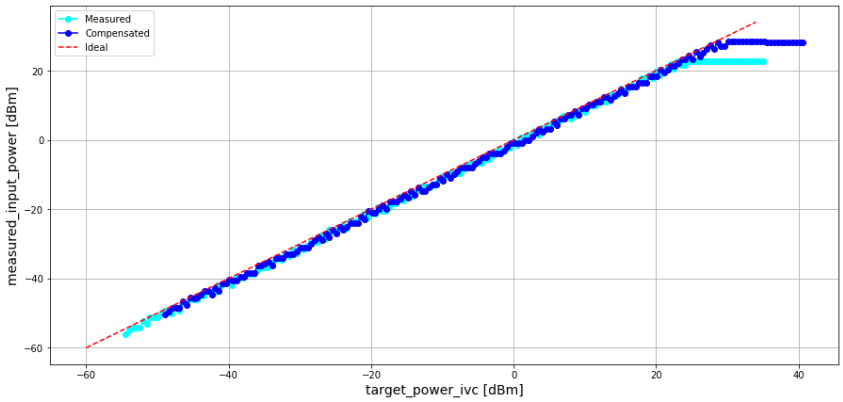


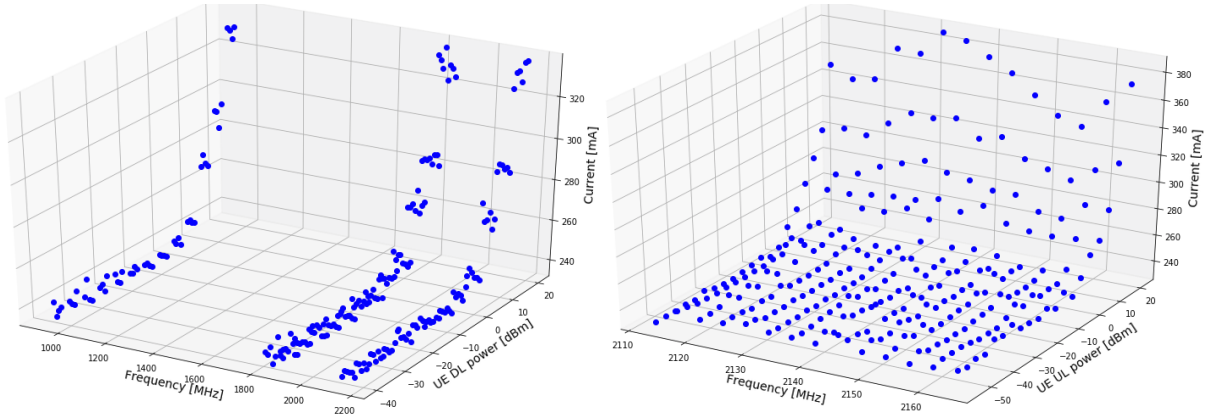
Fig. 5.21 Visualization of output of output power TC in MATRIX

Figure 5.21 shows a fine slope where all automated measurements allow to certify that the output power emitted by the IVC's RF antenna works as intended. To obtain this assessment I precisely measured the path attenuation caused by connectors and cable. It took a value of 5.2 dB in average due to the 3 dB power splitter that was initially conceived for cell handover while in two comm tester mode. From Figure 5.21, in turquoise, we see the real measured power at the comm tester's input RF connector –see Figure C.4 in Appendix C.1. I made the comm tester to compensate this path attenuation, as shown in blue. The maximum output power is 25 dBm. Above that, the power remains flat. On the contrary, RF link is out of synchronization below -73 dBm –Radio Resource Control (RRC) connection lost.

[7] Amaury –SIT Manager, confirmed this hypothesis. They update and synchronize all servers when no one is at the office, typically at odd hours. This shows the potential value of my work: I find problems which I escalate to the experts so they can devise why they happen.

RE 2.3: consumption

A study of IVC's consumption, in mA, is carried out based on its frequency and its RF antenna output power. The idea is to perform a frequency sweep all over the IVC's 3 compatible Universal Mobile Telecommunications System (UMTS) frequency bands –900, 1800 and 2100 MHz.



(a) (b)

Fig. 5.22 Visualization of output of "Frequency vs power vs consumption" TC in MATRIX: (a) all UMTS bands, (b) only band 1

Figure 5.22 shows two different TC realizations. All 3 bands are considered in Figure 5.22a. On the contrary, Figure 5.22b includes only the 2100 Mhz band number 1. As expected, the higher the IVC's RF antenna output power, the higher the battery consumption is. Even though it remains flat for a wide power range from -40 to 10 dBm, it rises exponentially beyond 10. It may suggest to ideally operate below 10 dBm.

5.2.3 Results and discussion

Once again, results present the IVC as a robust system. RE 2.1 showed possible reliability issues on the 0,000063% of iterations, or 1 every 150. We reported it to the Groupe Renault PnP experts at RSWL. For that, I arranged a meeting on the 12/06/2019 with different field-experts: a Software Test Architect, a PnP Software Architect, and a PnP Software System Engineer. During this internal RSWL meeting we agreed to the following terms:

1. The power linearity test, although interesting, does not correspond to RSWL. It is the main supplier, or Tier 1, who must certificate their furnished hardware components –IVC's Qualcomm RF modem in this case. However, it may be useful from a reliability testing point of view.

2. Although the connection time test is a meaningful test, we must ensure several factors first, including: (1) check the private network internal connections with Taoufik KEBBACHE (IT IS Security Manager IS at RSWL-Toulouse site), (2) verify availability of target IP address, and (3) retrieve CMW500 DAU logs in order to verify that it is the IVC the one causing errors and not someone else like the internal DAU, the internal private network, and so forth. However, the

interest related to this test case highly increases when dealing with a full IVC-to-AVNEXT link connection.

3. Power consumption tests and consumption validation must be carried out by the main supplier. However, it may serve as a basis for future test case development at a PnP team level. Alternative power test cases will be considered. For example, Multi-System consumption measurements.

5.3 Documenting my work

Even though I found some interesting results, the work is not done yet. It is well known that the "Manifesto for Agile Software Development" advocates for "Working software over comprehensive documentation" [26]. However, I successfully implemented and then analysed what was initially demanded. Thus, there is a need to make it easily reproducible. It is why I am currently developing a comprehensive guide in Confluence, the wiki at an Alliance level and very much used by any member at RSWL. It may help others to easily learn about what was achieved. Besides, it may motivate them to eventually obtain further, more complete analysis and results.

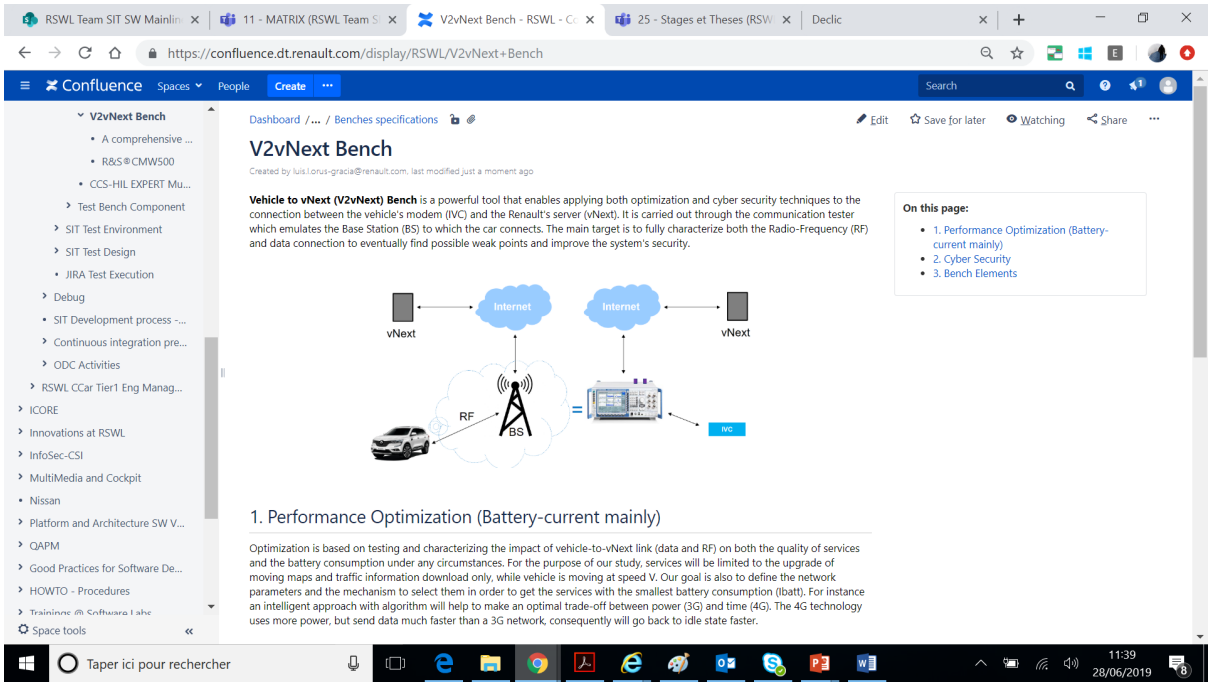


Fig. 5.23 Confluence documentation. "V2vNext" stands for IVC-to-VNEXT

Figure 5.23 shows a screenshot of what I am currently developing. It means the completion of my work on this project, ending the cycle: requirements, implementation, automation and, last but not least, documentation.

Chapter 6

Conclusions and future development

6.1 Conclusions

From a technical point of view, this project got two main achievements. On the one hand, I successfully demonstrated that cybersecurity tests can be fully automated and integrated into the MATRIX project. I boosted the relationships between the Groupe Renault cybersecurity experts at TCR and test automation experts in RSWL, providing a basis for future cooperative projects. On the other hand, I provided a solid automated test bench that laid the groundwork for future test cases combining cybersecurity algorithms with physical measurements and adaptable scenario settings. A simple TC enabling data gathering and further data science analysis will allow cybersecurity experts to develop broader security measures and expand the "secure by design" paradigm. Thanks to automated TC we are able to easily reproduce our test bench anywhere and collect data anytime. As previously stated: connected means data exchange. Data means information. Information is knowledge, and knowledge is power.

All in all, 8 different TC were presented: 5 related to cybersecurity and 3 related to performance. Despite being simple, they serve as basis for upcoming, more complex TC. They were a necessary step required before going into deeper detail and representing the foundation stone into a bigger project that is a completely validated, secure connected vehicle. All code I developed into this project will be accessible anywhere within the Alliance, in France and overseas. Near-shore[1] and off-shore [2].

From a practical point of view, this project reinforces the idea of cooperative work. I followed the previous results achieved by Xavier, ancient intern at RSWL, with regards to cybersecurity. He made a comprehensive and complete work in which he precisely described many concerns about computer security. He provided fundamental stages in embedded software cybersecurity certification. More precisely, he completely analyzed the whole IVC-to-VNEXT link in order to provide a formal validation tool focused on TLS and its possible weak points. I came later on and took advantage of what he made. It allowed me to partially cover some of his findings with my automated test cases. Despite being "just"

[1] European subsidiary software development centers located in France or Romania for example
[2] Asian subsidiary software development centers located in China or India for example

interns, we provided **added value** to the RSWL Intellectual Property (IP) and enjoyed from a mutual beneficial outcome.

6.2 Future development

I believe two different approaches can be followed up in terms of "future development". On the one hand we have the algorithms and analysis that enable the continuation of my work. Take for example the test case results related to the prediction of time peaks in IVC's . Even though we introduced a simple yet powerful analysis on Chapter 5, I lacked both data and time to go beyond. I would have had to measure and gather much more information in order to provide the predictive ML model with meaningful data to learn. For example, I could have used the daytime, the hour, the target ping address, the battery consumption, the number of running process simultaneously on the IVC, to name but a few, to make a more robust predictive model. Besides, this analysis may not be relevant in terms of this particular given problem. However, it may be used when studying anomaly detection or monitoring possible cyber attacks, enabling for more powerful systems built from different ideas, projects and developers. On the other hand, more conceptual ideas are likely to be continued. They are tunnelling, speed of vehicle, and cell handover. These are all 3 main aspects that were not unfortunately covered within this project. Even though I partially modelled adaptive path attenuation simulating tunneling and car movement, I did not finally used on the results. Besides, I initially tried to carry out cell handover between two different comm testers, with unpleasant results. Even though I achieved it several times –by lowering the power from one comm tester and rising the other one smoothly at the same time by hand, it was clear that a comm tester controller was indeed required –we can find the "RS® CMWC" on the lab. Just like in the real world, two different BS have a communications and synchronization protocol that enables the smooth change from one mobile cell to another without any loss of data connection. They may also have different frequencies so synchronization is key. However, I did it by hand and no synchronism measures were taken into consideration –more precisely, they were not even available. Therefore, cell handover remains an important role to cover in future development. Both tunneling and cell handover provide a full new world of possibility in regards cybersecurity attacks. It is where physical and environmental conditions that are not ideal result into failures and malfunctions –such as the well-known "Sybil Attacks" [45] that I initially studied but never implemented.

Apart from physical configurations and further development, more complex cybersecurity attacks and defenses may be implemented. I did not try to break the TLS connection by implementing already well-known attacks –as shown in Appendix B.3. MITM attacks are also available, for I already set up the PC 2, or packet sniffer, in the middle of the connection. Alternatively, more data science techniques may be applied to measurements where applicable. For example, they may allow us to monitor and detect anomalies where power consumption exceptionally rises for no reason.

6.3 Personal review

From a **professional point of view** I feel recognition and a feeling of pride. I was given the opportunity to deal with real-world systems –and real-world problems, and found to be successful. From

minute one I was in full charge of the test bench automation. Once I demonstrated to my manager I was a responsible professional, I was left alone with the comm tester, a device that goes up to 400k USD when new. I also attended MATRIX daily meetings and felt better than welcome –for I was fully immersed into a "Scrum" (an **Agile framework**) and a daily stand-up with other team developers and interns. I was given the opportunity to listen to engineers but also to be heard, receiving supportive and encouraging messages in a weekly basis. Even though, as assistant, did not belong to any specific "Sprint", I constantly worked on my own, personalized "Epics". MATRIX and the connected vehicles are a complex topic that require an adaptive process. There are many things involved to consider, so learning is key. I was given the time and materials to keep up with my global understanding of the system, and for that I am thankful. Besides, Dominique, my advisor at Renault Software Labs (RSWL), gave me reasonable freedom in terms of the project evolution so I could choose where I wanted to invest my time and effort, collaborating on the things that I liked more. From a "big picture" point of view, I really value the fact that I helped somehow developing –in a wider sense, a better, more secure connected vehicle. I really learned how to work in a complex project involving many people from many different backgrounds and fields of expertise.

Before this project, I also successfully completed an internship on data science focusing on the automated generation of critical test cases in the field of ADAS. This means that over 90% of my current professional experience applies to RSWL. Therefore I can be no wrong when I say that the Groupe Renault in general and RSWL in particular bring professional opportunities to young graduates and students. They offer full support to engineering students, and for that I am thankful.

From a **personal point of view** I cannot thank enough both the Universitat Politècnica de Catalunya, or Technical University of Catalonia, in English (UPC), and Institut Supérieur de l'Aéronautique et de l'Espace, or Higher Institute of Aeronautics and Space, in English (ISAE-SUPAERO), Engineering Schools. They allowed me to fully immerse into a different country. Thanks to them, I call France now a home. Not only I enjoyed from an international Double Degree program, but also enjoyed from a dual student-employee status at Renault Software Labs (RSWL).

It is only by investing on the younger generations how we push Europe forward and build a better future for everyone of us all.

References

[1] L. Laurent for Bloomberg at The Washington Post. «Apple, Facebook and Google Have Lost the Monopoly Argument», 2019.

[2] Society of Automotive Engineers (SAE). «Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles», 2018.

[3] European Telecommunications Standards Institute (ETSI). «5G» https://www.etsi.org/technologies/5g, 2019.

[4] Association For Safe International Road Travel. «Annual Global Road Crash Statistics», 2018.

[5] Euro NCAP. «Euro NCAP 2025 Roadmap», 2017.

[6] Perkinscoie. «Autonomous Vehicles Survey Report», 2019.

[7] Foley. «Connected Cars Autonomous Vehicles Survey», 2017.

[8] X. Guerin. «FOTA & Remote Services Security Validation Internship», 2018.

[9] Groupe Renault. «Groupe Renault: Facts & Figures», 2018.

[10] Groupe Renault. «Worldwide Sales Reults 2018: Groupe Renault Sales Reached 3.9 Million Vehicles, up 3.2% with Jinbei and Huasong» (Press release), 2019.

[11] The Alliance. «About us» (https://www.alliance-2022.com/about-us/), 2019.

[12] Oxford University Press. «Lexico» (https://www.lexico.com/en), 2019.

[13] Groupe Renault. «Renault Internal», 2019.

[14] engadget.com. «Tesla will soon downgrade software on the entry-level Model 3» (https://www.engadget.com/2019/06/08/tesla-model-3-software-downgrade/), 2019.

[15] ACEA. «Average Vehicle Age», 2017.

[16] The Alliance. «Connected vehicles» (https://www.alliance-2022.com/connected-vehicles/), 2019.

[17] K. Lindemann for Elektrobit. «Next Generation of IVI Systems: Android Automotive», 2018.

[18] Altran. «Cybersecurity in Automotive: How to Start Ahead of Cyber Threats?», 2018.

[19] S. Holtmanns, S.P. Rao and I. Oliver. «User Location Tracking Attacks for LTE Networks Using the Interworking Functionality» (IFIP Networking), 2016.

[20] P. Ficheux, J. Rosen, and V. Dehors for Smile. «Sécurité des Objets Connectés», 2019.

[21] International Organization for Standardization (ISO). «ISO/IEC 27000 family: Information security management systems», 2013.

[22] International Organization for Standardization (ISO). «ISO/IEC 26262 family: Road vehicles - Functional safety», 2011.

[23] IBM. «Artificial intelligence for a smarket kind of cybersecurity», 2019.

[24] L. Torvalds. «Git» (https://git-scm.com/), 2005.

[25] M. Fowler. «Continuous Integration», 2006.

[26] W. Cunningham. «Manifesto for Agile Software Development» (https://agilemanifesto.org/), 2001.

[27] A.C. Hardy. «Agile Team Organisation: Squads, Chapters, Tribes and Guilds» (https://medium.com/@achardypm/agile-team-organisation-squads-chapters-tribes-and-guilds-80932aceofdc), 2016.

[28] Statista. «Global smart phone OS market share held by RIM (BlackBerry) from 2007 to 2016», 2019.

[29] Statista. «Mobile phone vendor's market share in sold units to end users worldwide from 1997 to 2014», 2019.

[30] Scrum Alliance. «What is Scrum? An Agile Framework for Completing Complex Projects», 2016.

[31] J. Humble. «Continuous Delivery vs Continuous Deployment», 2010.

[32] ISO/IEC/IEEE. «29119 Software Testing» (http://www.softwaretestingstandard.org/), 2013.

[33] P. Klärck, J. Härkönen et al. «Robot Framework» (https://robotframework.org/), 2008.

[34] International Software Testing Qualifications Board (https://www.istqb.org/).

[35] Deloitte. «Global mobile consumer trends, 2nd edition», 2017.

[36] Trustonic. «Advanced automotive security solutions – Protection for mobile apps, digital keys, vehicle electronics, in-car entertainment (ICE) in-vehicle infotainment (IVI)», 2019.

[37] Project Jupyter. «Jupyter Notebook» (https://jupyter.org/), 2015.

[38] Internet Engineering Task Force (IETF). «User Datagram Protocol (UDP)» (https://tools.ietf.org/html/rfc768), 1980.

[39] M. Rouse for TechTarget. «What is UDP/What is TCP», 2018 & 2014.

[40] Internet Engineering Task Force (IETF). «Transmission Control Protocol (TCP)» (https://tools.ietf.org/html/rfc793), 1981.

[41] P. Biondi. «Scapy: Packet crafting for Python2 and Python3» (https://scapy.net/), 2005.

[42] E. Perraud. «Machine Learning Algorithms of Detection of DoS Attacks on an Automotive Telematic Unit» (IJCNC), 2019.

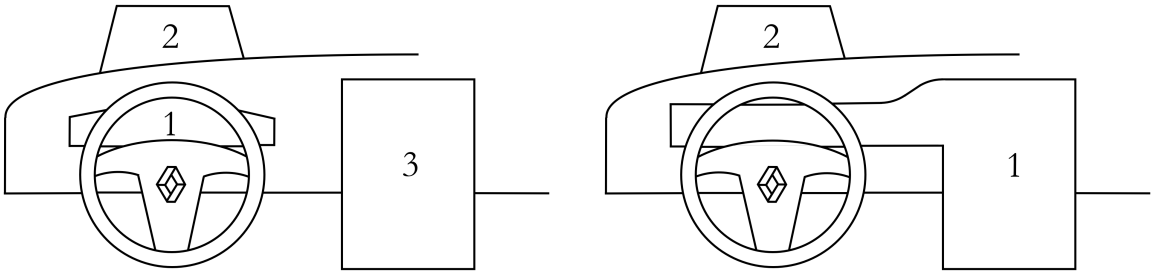
[43] J. Shlens. «A Tutorial on Principal Component Analysis: Derivation, Discussion and Singular Value Decomposition», 2003.

[44] N. Gugulothu, V. TV, P. Malhotra, et al. «Predicting Remaining Useful Life using Time Series Embeddings based on Recurrent Neural Networks» (arXiv), 2017.

[45] M.S. Naveed, Dr. M.H. Islam. «Detection of Sybil Attacks in Vehicular Ad hoc Networks Based on Road Side Unit Support» (IJSER), 2015.

Appendix A

Added value



(a) (b)

Fig. A.1 Upcoming cockpit architecture: (a) 3-display cockpit, (b) L-shape 2-display display cockpit

Connected vehicles will bring added value to car makers. We live in a digital society now where screens grow in size as time goes by. One of the features that increases a car's added value is that of cockpit display surface, as seen in Figure x.a. It is formed of three major displays: the cluster (1), located behind the steering wheel and displaying the dashboard and car parameters such as fuel consumption or current speed, Head-Up Display (HUD) (2), a mirrored display located straight into the windshield providing traffic information without perturbing the driver's attention, and Center Information Display (CID) (3), regarded as the infotainment hub, located between the driver and the front-seat passenger, improving the driving experience. Some current market values of such cockpit displace surface:

- Tesla Model S: 1100 cm2
- Audi A8: 980 cm2
- Mercedes E class: 810 cm2
- Volkswagen Passat: 730 cm2
- New Renault Clio V: 415 cm2

It is worth noting that smartphone screens present higher PPI values with respect to vehicle screens as smaller viewing distances are required. Besides, elderly people need higher values given their vision

quality decreases over the years. In other words, same viewing distance require different PPI values based on the driver's age. Current vehicle screens have a minimum of 100 PPI for a maximum viewing distance of 85 cm. Comparatively speaking, a high-end Samsung screen bears over 500 PPI requiring less than 25 cm of viewing distance [13].

At an Alliance level, Renault chose to use its CID in portrait mode whereas Nissan uses it in landscape mode. Key features are therefore based upon the increasing size of the triplet cluster/HUD/CID, off-board content and settings, and skin updates. Infotainment systems must be both emotional and functional. It means they must be stylish and bear a nice appearance while being easy to read, accessible, user friendly, and, last but not least, safe in use. From a cost analysis perspective, there is also a trade-off between HMI customer value estimation and its actual cost. For example, an Audi A4 system cost ranges from 200 to 400 € but implies an added value from 300 to 700 € respectively. A normal evolution in terms of quality goes from a low variant system without any screen, lacking the triplet cluster/HUD/CID, to a standard system with both small cluster/CID screens, an upper system with larger cluster/CID screens, and finally a high-end system with even larger cluster/CID screens of around 12 inches each and a HUD. It is believed that eventually, as shown in Figure x.b, we will have a L-shape (1) cockpit screen including all previous luxury-appearance systems, plus the already mentioned HUD (2).

Appendix B

Transport Layer Security (TLS)

B.1 Secure connections

The TLS protocol, or Transport Layer Security, is the most important and widespread solution for any secure exchange over the network. It was conceived to replace SSL, its now-deprecated predecessor. Based on the peer-to-peer paradigm, it encapsulates application data to provide both integrity and authentication. According to the Oxford Dictionary, integrity is "the state of being whole and undivided". More precisely, the "internal consistency or lack of corruption in electronic data". Similarly, authentication is "the process or action of proving or showing something to be true, genuine, or valid". Let's suppose that Alice sends data to Bob over the Internet. In plain English, TLS will assure Bob that Alice is really Alice, and that nobody altered what she just sent him.

B.2 Versions

To date, four different versions have been officially released:

- TLS 1.0, first defined in RFC 2246 in 1999, was conceived as an upgrade of SSL 3.0, vulnerable to the POODLE attack. SSL 3.0 was deprecated in RFC 7568 in 2015. They are not compatible (see RFC 6176). A TLS 1.0 connection can be downgraded to SSL 3.0, making it insecure and not recommended for use. Thus, its deprecation is planned for 2020.
- TLS 1.1, first defined in RFC 4346 in 2006, was conceived as un upgrade to TLS 1.0, vulnerable to CBC and BEAST attacks. Some of the major technological companies have planned to deprecate it in 2020.
- TLS 1.2, first defined in RFC 5246 in 2008, was conceived as un upgrade to TLS 1.1.
- TLS 1.3, first defined in RFC 8446 in 2018, was conceived as un upgrade to TLS 1.2. It is the most recent version of TLS, and it is advisory to use it for assured protection.

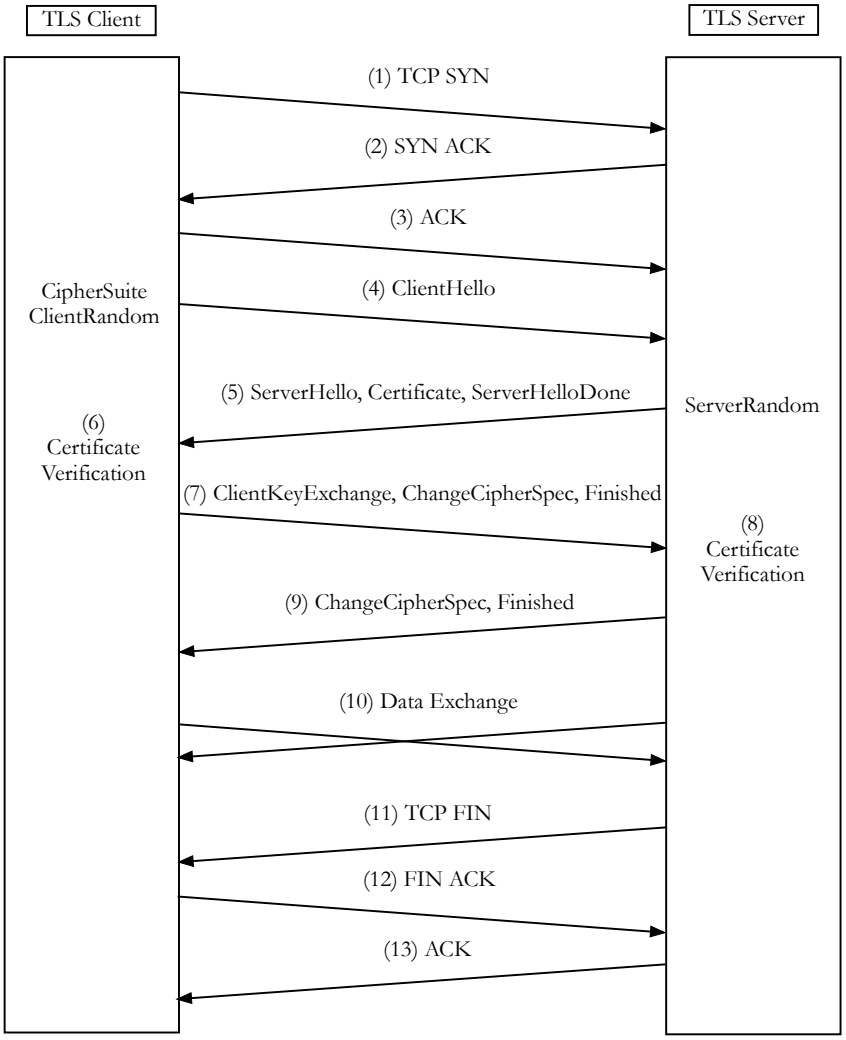


Fig. B.1 Fully labeled TLS handshake and conversation

B.3 Vulnerabilities

Most of the TLS vulnerabilities and attacks are based on downgrading the current TLS version to the oldest one possible, ideally deprecated, in order to exploit known vulnerabilities. As countermeasures, there are many sites that only accept TLS 1.3 as legit. Other famous TLS attacks are CRIME, BREACH, and Heartbleed. We present on Table B.1 the actions to be taken in order to avoid the aforementioned attacks [8].

Attack	Description	Countermeasures
POODLE (CVE-2014-3566)*	MITM attack	Disable SSL 2.0, SSL 3.0 Use only TLS 1.1 or higher Use TLS FALLBACK SCSV cipher
BEAST (CVE-2011-3389)	Cyber attack on CBC cipher used by SSL 3.0 and TLS 1.0	Use only TLS 1.1 or higher
CRIME (CVE-2012-4929)	Use DEFLATE compression method to brute force data as cookies	DEFLATE not included
BREACH (CVE-2013-3587)	Use the HTTP compression algorithm	Disable HTTP compression, separate secrets from user input, mask and randomize them, protect vulnerable pages with CSRF, rate-limiting requests
Heartbleed (CVE-2014-0160)	Improper input validation, buffer over-read	Use OpenSSL 1.0.1g or higher
DROWN (CVE-2015-3197)	Able to fully decrypt a TLS session	Disable SSL 2.0 Use OpenSSL 1.0.2f or higher
LOGJAM (CVE-2015-4000)	Insecure Diffie-Hellman (DH) key exchange with key < 2048 bits	Use a DH key higher than 2048 bits
Secure Renegotiation (CVE-2009-0160)	Vulnerability on the renegotiation process initiated when the client is setting a new key	Enable secure renegotiation by preventing the client to initiate it

(*) Common Vulnerabilities and Exposures, a database of publicly known cyber attacks

Table B.1 TLS attacks, descriptions, and countermeasures

Appendix C

Setup

C.1 Figures

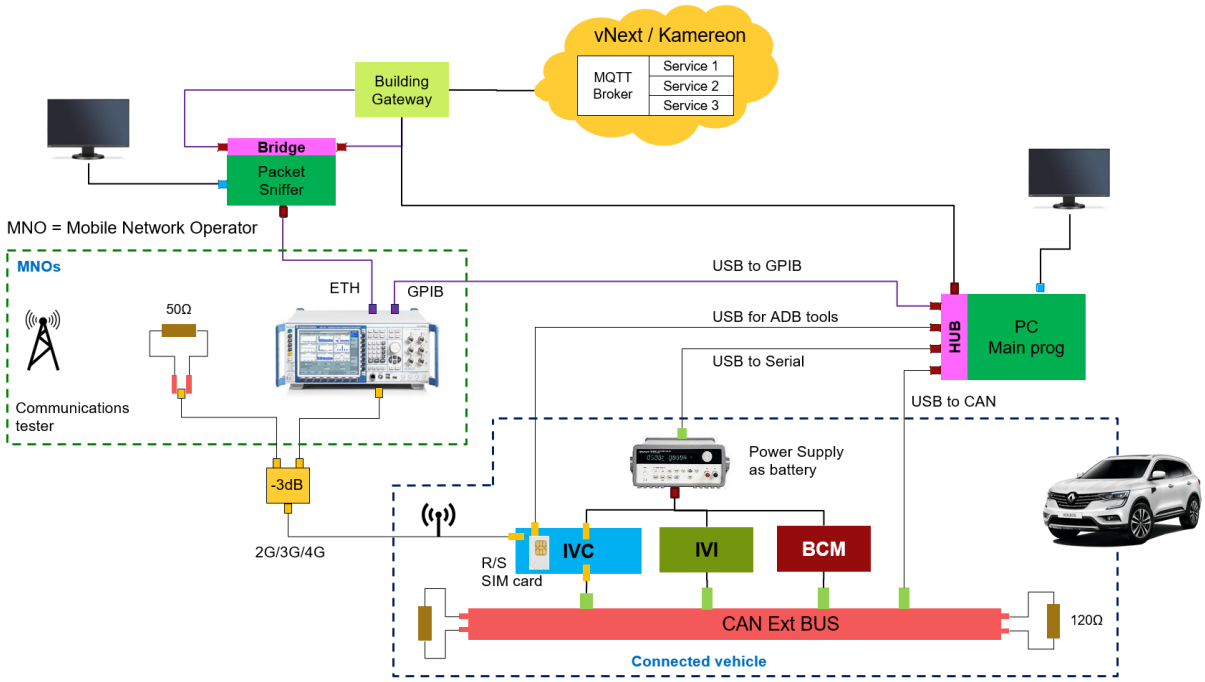


Fig. C.1 Test bench setup. Only one comm tester was finally used (see 50Ω resistor)

Fig. C.2 Laboratory setup

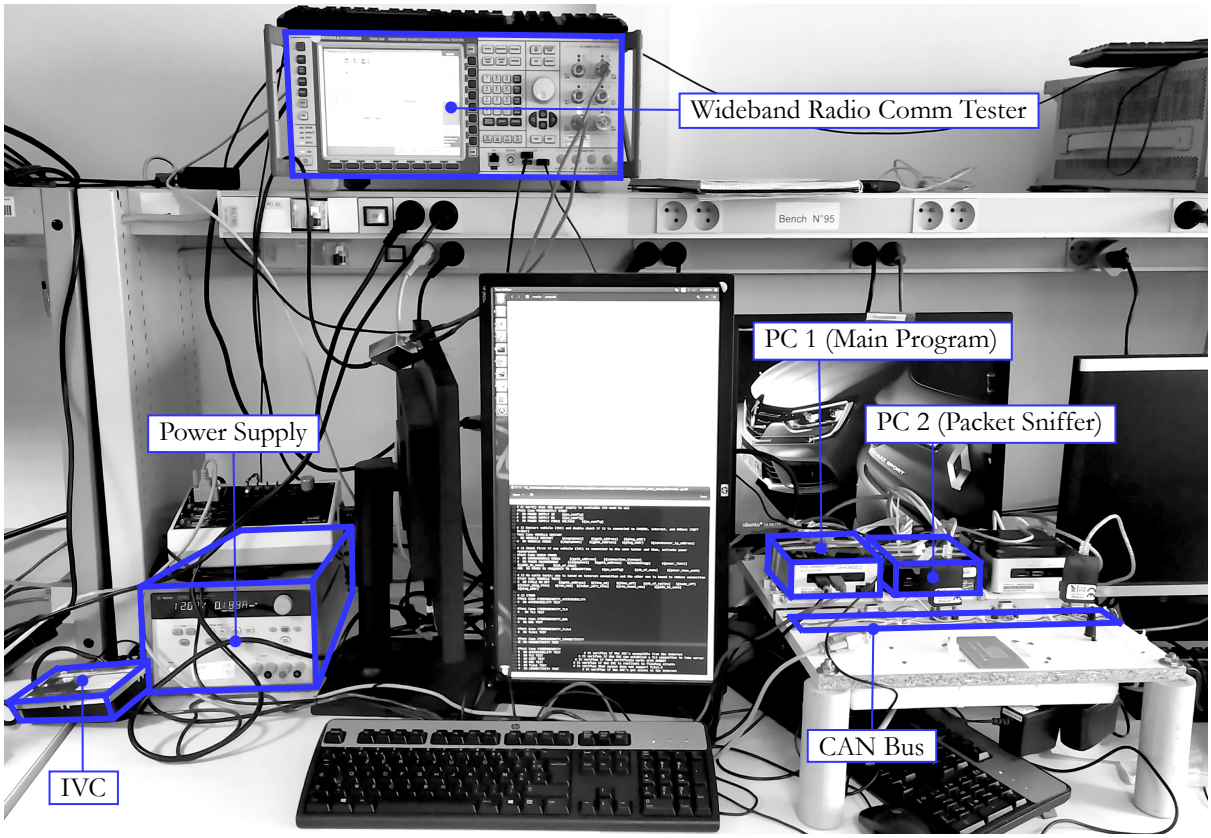


Fig. C.3 Laboratory setup with labels

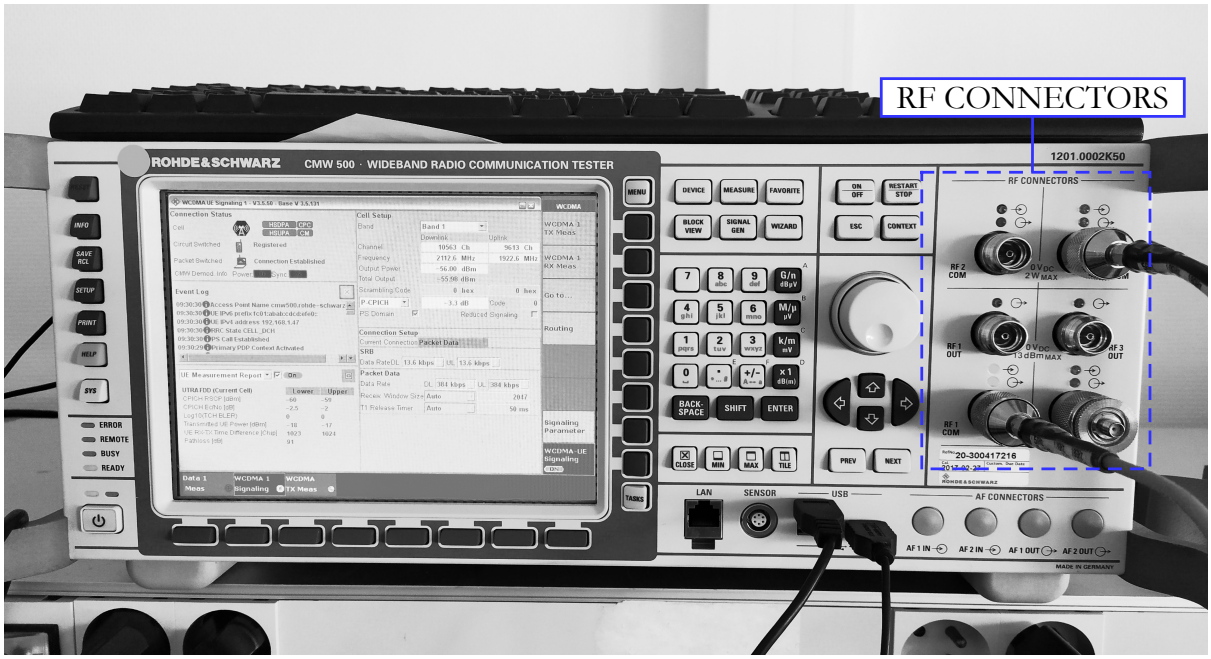


Fig. C.4 Comm tester's RF connectors. We only used port "RF 1 COM" during this project

Fig. C.5 Comm tester front display. Example of IVC's Connection ESTablished (CEST)

C.2 Specifications: standards and entities

IEEE 488 (GPIB)	
Description	Specification for short-range digital communications
RS-232 (Serial)	
Description	Standard for serial communication
Hardware	UART, computer hardware device for asynchronous serial communications
SCPI	
Description	Standard for syntax and commands to use in controlling programmable test and measurement devices
Transmitted over	GPIB, Serial
CAN Bus	
Description	Vehicle bus standard to allow inter-microcontroller communications
USB	
Description	Inter-computer and inter-device communication standard

Table C.1 Standards used on the test bench

Wideband Radio Communication Tester	**2G/3G/4G Base Station**
Maker	Rohde&Schwarz GmbH & Co KG
Name	R&S® CMW500
Model	1201.0002K50, selection R&S®CMW-S600B
Software version	3.5.131
IVC	**Car's modem**
Maker	Continental Automotive Systems, Inc.
Name	AIVC Board
Software version	12.1
OpenSSL version	1.1.0, 25 Aug 2016
Power Supply	**Car's battery**
Maker	Agilent Technologies, Inc.
Name	Agilent E3648A
PC 1	**Running MATRIX**
Maker	Intel Corporation
Name & Processor	NUC, Intel®Core i7-5557U CPU @ 3.10GHz x 4
OS	Ubuntu 16.04 LTS
OpenSSL	1.1.1a, 20 Nov 2018
PC 2	**Running Wireshark**
Maker	Intel Corporation
Name & Processor	NUC, Intel®Core i5-3427U CPU @ 1.80GHz x 4
OS	Ubuntu 16.04 LTS

Table C.2 Entities included within the test bench

Appendix D

Code

D.1 eq_cmw500_test.robot

```
#
# 2018 Developed by Renault SW Labs,
# an affiliate of RENAULT s.a.s. which holds all intellectual property rights.
# Use of this software is subject to a specific license granted by Renault s.a.s.
#
*** Settings ***
Library           eq_cmw500_test.eq_cmw500.CMW500    use_grpc=FALSE

*** Keywords ***
CHECKSET GPIB STATUS
  [arguments]    ${equipment}    ${gpib_address}
  Log To Console    Checking if commtester:${equipment} is available on GPIB address:${gpib_address}
  check if gpib available    ${equipment}    ${gpib_address}

SET COMMTESTER CONFIG
  [arguments]    ${equipment}    ${gpib_address}    ${technology}
  Log To Console    Configuration:${technology} from YAML file will be loaded into the commtester:${equipment}
  set configuration    ${equipment}    ${gpib_address}    ${technology}

DO COMMTESTER RECALL
  [arguments]    ${equipment}    ${gpib_address}    ${file_to_recall}
  Log To Console    Found a previous file to recall. It will be loaded.
  load recall configuration    ${equipment}    ${gpib_address}    ${file_to_recall}

CHECK INTERNET CONNECTIVITY
  [arguments]    ${equipment}    ${gpib_address}    ${ping_addr}    ${ct_ping_timeout}
  Log To Console    Checking Internet connectivity: will try PING from commtester:${gpib_address} to external ip address:${
     ping_addr}
  check internet connection    ${equipment}    ${gpib_address}    ${ping_addr}    ${ct_ping_timeout}

CHECKSET GO TO LOCAL
  [arguments]    ${equipment}    ${gpib_address}
  Log To Console    Returning control of commtester:${equipment} to end user in local
  return control to end user    ${equipment}    ${gpib_address}

CHECK COMMTERSTER SOFTWARE VERSION
  [arguments]    ${equipment}    ${gpib_address}
  Log To Console    Checking current software version in commtester:${equipment}
  verify valid version    ${equipment}    ${gpib_address}

CHECKSET WCDMA OFF
  [arguments]    ${gpib_address}
  Log To Console    Shutting SIG 1 WCDMA down if any in RF 1 COM from comm tester
  ${result} =    shut down rf    ${gpib_address}
  [Return]    ${result}

CHECKSET WCDMA ON
  [arguments]    ${gpib_address}
  Log To Console    Starting SIG 1 WCDMA in RF 1 COM from comm tester
  ${result} =    start rf    ${gpib_address}
  [Return]    ${result}

CHECKSET CONNECTION CEST
  [arguments]    ${gpib_address}    ${cest_timeout}    ${cest_interval_in_s}
  Log To Console    Checking if the IVC's state is CEST into the comm tester for a specific given timeout
  ${result} =    check if ivc is cest    ${gpib_address}    ${cest_timeout}    ${cest_interval_in_s}
  [Return]    ${result}

CHECKSET COMM TESTER IP ADDRESS
```

```
  [arguments]    ${gpib_address}    ${commtester_ip_address}
  Log To Console    Verifying if comm tester's provided ip address is correct
  check if comm tester ip address is valid        ${gpib_address}    ${commtester_ip_address}

DEACTIVATE DNS
  [arguments]    ${equipment}    ${gpib_address}
  Log To Console    Shutting down DNS
  shut down dns    ${equipment}    ${gpib_address}

CHECKSET DNS
  [arguments]    ${equipment}    ${gpib_address}    ${dns_target_address}    ${dns_timeout}
  Log To Console    Starting the comm tester DNS, and restarting if necessary
  start and test dns    ${equipment}    ${gpib_address}    ${dns_target_address}    ${dns_timeout}

CHECKSET REBOOT DAU
  [arguments]    ${equipment}    ${gpib_address}    ${wait_timer}
  Log To Console    Booting DAU: Data Application Unit
  reboot dau    ${equipment}    ${gpib_address}    ${wait_timer}

CHECKSET UPDATE SCREEN
  [arguments]    ${equipment}    ${gpib_address}
  Log To Console    COmm tester:${equipment} display will be kept alive
  keep display alive    ${equipment}    ${gpib_address}

CHECKSET POWER MEASUREMENT
  [arguments]    ${equipment}    ${gpib_address}    ${power_limit}
  Log To Console    Activate IVC's output power measurement from comm tester
  ${result} =    activate power measurement    ${equipment}    ${gpib_address}    ${power_limit}
  [Return]    ${result}

DO POWER MEASUREMENT
  [arguments]    ${equipment}    ${gpib_address}    ${target_power}
  #Log To Console    Measuring comm tester's input power
  ${result} =    carry out power measurement    ${equipment}    ${gpib_address}    ${target_power}
  [Return]    ${result}

SET POWER VALUE
  [arguments]    ${equipment}    ${gpib_address}    ${target_power}
  #Log To Console    Sending an output power order of ${target_power} dBm to the IVC
  set power to the ivc    ${equipment}    ${gpib_address}    ${target_power}

CHECKSET ACTIVATE POWER SUPPLY
  [arguments]    ${yml_path}    ${pw_config_name}
  activate power supply    ${yml_path}    ${pw_config_name}

DO CURRENT MEASUREMENT
  [arguments]    ${yml_path}    ${pw_config_name}
  ${result} =    carry out current measurement    ${yml_path}    ${pw_config_name}
  [Return]    ${result}

SET FREQUENCY VALUE
  [arguments]    ${gpib_address}    ${operation_band}    ${dl_frequency}    ${ul_frequency}
  set specific frequency to the ivc    ${gpib_address}    ${operation_band}    ${dl_frequency}    ${ul_frequency}
```

D.2 udp_flood.py

```
#
# 2016 Developed by Ananasr (https://gist.github.com/Ananasr) and
# adapted/modified by L Orus on 2018. UDP flood attack.
#
import time, socket, random, sys

def usage():
    print("Usage: " + sys.argv[0] + " <ip> <port> <second>")

def flood(victim, vport, duration):
    # creation of server: "SOCK_DGRAM" = UDP type program, 1024 = 1B
    client = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    bytes = random._urandom(1024)
    timeout = time.time() + duration
    sent = 0
    print("-"*40)
    while 1:
        if time.time() > timeout:
            break
        else:
            pass
        client.sendto(bytes, (victim, vport))
        sent = sent + 1
        print("[Attack] %s ---> %s:%s"%(sent, victim, vport), end='\r')

def main():
    print(len(sys.argv))
    if len(sys.argv) != 4:
        usage()
    else:
        flood(sys.argv[1], int(sys.argv[2]), float(sys.argv[3]))

if __name__ == '__main__':
    main()
```

D.3 TC_SET_DEFAULT_CONFIG_CMW500.robot

```
#
# 2019 Developed by Renault SW Labs,
# an affiliate of RENAULT s.a.s. which holds all intellectual property rights.
# Use of this software is subject to a specific license granted by Renault s.a.s.
#
*** Settings ***
Documentation     Test to load the by-default settings in the CMW500 so we can run TC properly
...
...               It will the initial settings to the CMW500 and will verify its connection to the Internet. If necessary,
    Linux drivers will be installed.
...
Test Setup        SETUP_TC_SET_DEFAULT_CONFIG_CMW500
Test Teardown     TEARDOWN_TC_SET_DEFAULT_CONFIG_CMW500
Resource          ../HLK/eq_cmw500.robot

*** Variables ***
${equipment}              CMW500
${gpib_address}           20
${technology}             default_3G
${file_to_recall}         d:\\Rohde-Schwarz\\CMW\\Data\\Save\\az02476\\cmw5002configleft.dfl
${ping_addr}              k.ro
${ct_ping_timeout}        30
${dns_target_address}     www.google.es
${dns_timeout}            3
${wait_timer}             80

*** Test Cases ***
TC_SET_DEFAULT_CONFIG_CMW500
    Log To Console    ${\n}${\n}***** Begin Test Execution *****${\n}
    CHECK COMMTERSTER SOFTWARE VERSION    ${equipment}    ${gpib_address}
    DO COMMTESTER RECALL    ${equipment}    ${gpib_address}    ${file_to_recall}
    SET COMMTESTER CONFIG    ${equipment}    ${gpib_address}    ${technology}
    CHECKSET DNS    ${equipment}    ${gpib_address}    ${dns_target_address}    ${dns_timeout}
    CHECK INTERNET CONNECTIVITY    ${equipment}    ${gpib_address}    ${ping_addr}    ${ct_ping_timeout}

*** Keywords ***
SETUP_TC_SET_DEFAULT_CONFIG_CMW500
    Log To Console    ${\n}***** Setup Initial Conditions *****${\n}
    CHECKSET GPIB STATUS    ${equipment}    ${gpib_address}

TEARDOWN_TC_SET_DEFAULT_CONFIG_CMW500
    Run Keyword And Ignore Error    Log To Console    ${\n}***** Teardown *****${\n}
    Run Keyword And Ignore Error    CHECKSET GO TO LOCAL    ${equipment}    ${gpib_address}
```

D.4 TC_CONNECT_IVC_TO_INTERNET.robot

```
#
# 2019 Developed by Renault SW Labs,
# an affiliate of RENAULT s.a.s. which holds all intellectual property rights.
# Use of this software is subject to a specific license granted by Renault s.a.s.
#
*** Settings ***
Documentation     Test to check if the IVC's got connection to the internet via the comm tester
...
...               It will initially check if there's RF connection to later check the Internet
...     To do so, it will access the comm tester via the GPIB command and retrieve its
...     RF interface IP address to check if it's same as input target IP
...
Test Setup        SETUP_TC_CONNECT_IVC_TO_INTERNET
Test Teardown     TEARDOWN_TC_CONNECT_IVC_TO_INTERNET
Resource          ../HLK/vehicle_cybersec.robot
Resource          ../HLK/eq_cmw500.robot

*** Variables ***
${gpib_address}             20
${commtester_ip_address}    192.168.1.16
${ping_addr}                k.ro
${activate_firewall}        TRUE
${cest_timeout}             240
${cest_interval_in_s}       1
${interface_rf}             rmnet_data5
${ct_timeout}               15
${can_to_deactivate}        slcan0
${inter_ping_time}          1
${internet_address}         k.ro
${equipment}                CMW500
${nb_of_sent_pings}         1
${wait_timer}               80
${dns_target_address}       www.google.es
${dns_timeout}              15

*** Test Cases ***
TC_CONNECT_IVC_TO_INTERNET
    Log To Console    ${\n}${\n}***** Begin Test Execution *****${\n}
    ${ct_result} =    CHECK IVC TO COMM TESTER    ${commtester_ip_address}    ${interface_rf}    ${ct_timeout}    ${
      inter_ping_time}    ${nb_of_sent_pings}
    ${internet_result} =    CHECK IVC TO INTERNET    ${internet_address}    ${interface_rf}    ${ct_timeout}    ${
      inter_ping_time}    ${nb_of_sent_pings}
    Log To Console    ct_result:${ct_result} --- internet_result:${internet_result}
    Run Keyword If    "${ct_result}" == "-1"    Log To Console     ct_result == -1 --- TRUE
    Run Keyword If    "${ct_result}" != "-1"    Log To Console     ct_result == -1 --- FALSE
    Run Keyword If    "${internet_result}" == "-1"    Log To Console     internet_result == -1 --- TRUE
    Run Keyword If    "${internet_result}" != "-1"    Log To Console     internet_result == -1 --- FALSE
    #Run Keyword If    "${ct_result}" == "-1" or "${internet_result}" == "-1" #Si pongo 'or', el Log To Console de despues no
       funciona: 0 input arguments...wtf?
    Run Keyword If    "${internet_result}" == "-1"
    ...    Run Keywords
    ...    Log To Console    ${\n}Ping did not work. Starting main approach to solve the problem${\n}
    ...    DEACTIVATE DNS    ${equipment}    ${gpib_address}
    ...    CHECKSET REBOOT DAU    ${equipment}    ${gpib_address}    ${wait_timer}
    ...    CHECKSET WCDMA ON    ${gpib_address}
    ...    CHECKSET DNS    ${equipment}    ${gpib_address}    ${dns_target_address}    ${dns_timeout}
    ...    CHECKSET CONNECTION CEST    ${gpib_address}    ${cest_timeout}    ${cest_interval_in_s}
    ...    CHECK IVC TO COMM TESTER    ${commtester_ip_address}    ${interface_rf}    ${ct_timeout}    ${inter_ping_time}
      ${nb_of_sent_pings}
    CHECKSET FIREWALL STATUS    ${interface_rf}

*** Keywords ***
SETUP_TC_CONNECT_IVC_TO_INTERNET
    Log To Console    ${\n}***** Setup Initial Conditions *****${\n}
    CHECKSET UPDATE SCREEN    ${equipment}    ${gpib_address}
    CHECKSET WCDMA OFF    ${gpib_address}
    CHECKSET DEACTIVATE CAN    ${can_to_deactivate}
    #DO CURRENT MEASUREMENT    ${yml_path}    ${pw_config_name}
    #CHECKSET ACTIVATE POWER SUPPLY    ${yml_path}    ${pw_config_name}
    CHECKSET ACTIVATE CAN
    CHECKSET IVC STATUS AND REBOOT
    CHECKSET WCDMA ON    ${gpib_address}
    CHECKSET CONNECTION CEST    ${gpib_address}    ${cest_timeout}    ${cest_interval_in_s}
    CHECKSET COMM TESTER IP ADDRESS    ${gpib_address}    ${commtester_ip_address}

TEARDOWN_TC_CONNECT_IVC_TO_INTERNET
    Run Keyword And Ignore Error    Log To Console    ${\n}***** Teardown *****${\n}
    Run Keyword And Ignore Error    CHECKSET GO TO LOCAL    ${equipment}    ${gpib_address}
```

D.5 TC_IVC_RELIABILITY_INTERNET.robot

```
#
# 2019 Developed by Renault SW Labs,
# an affiliate of RENAULT s.a.s. which holds all intellectual property rights.
# Use of this software is subject to a specific license granted by Renault s.a.s.
#
*** Settings ***
Documentation     Test to "test" the IVC's reliability
...
...               It will switch the comm tester RF ON/OFF in order to measure the time to reconnection to Internet
...
Test Setup        SETUP_TC_IVC_RELIABILITY_INTERNET
Test Teardown     TEARDOWN_TC_IVC_RELIABILITY_INTERNET
Resource          ../HLK/vehicle_cybersec.robot
Resource          ../HLK/eq_cmw500.robot
Library           DateTime

*** Variables ***
${gpib_address}             20
${internet_address}         k.ro
${interface_rf}             rmnet_data5
${ct_timeout}               30
${inter_ping_time}          1
${nb_of_sent_pings}         1
${cest_timeout}             180
${cest_interval_in_s}       1
${name_to_save}             results/results_tc_ivc_reliability_internet
${equipment}                CMW500
${number_of_iterations}     6000
${type_of_file}             ping
${can_to_deactivate}        slcan0
${zero}                     0
${result}                   0
${error_log}                ok

*** Test Cases ***
TC_IVC_RELIABILITY_INTERNET
    Log To Console    ${\n}${\n}***** Begin Test Execution *****${\n}
    ${initial_date} =    Get Current Date  result_format=%Y-%m-%d %H:%M:%S.%f
    ${initial_time} =    Convert Date  ${initial_date}  datetime
    CREATE NEW FILE    ${name_to_save}    ${initial_time}    ${type_of_file}
    :FOR    ${i}    IN RANGE    9999
    \    Exit For Loop If    ${i} == ${number_of_iterations}
    \    ${iteration_index} =    Evaluate    ${i} + 1
    \    ${rf_off_result} =    CHECKSET WCDMA OFF    ${gpib_address}
    \    Log To Console    rf_off_result:${rf_off_result}
    \    Run Keyword If    "${rf_off_result}" == "False"    CHECKSET GPIB STATUS    ${equipment}    ${gpib_address}
    \    CHECKSET WCDMA ON    ${gpib_address}
    \    ${error_log}    Set Variable If    "${rf_off_result}" == "False"    ko: gpib connection lost > reboot drivers    ok
    \    Log To Console    error_log:${error_log}
    \    ${current_date_before} =    Get Current Date  result_format=%Y-%m-%d %H:%M:%S.%f
    \    ${time_before} =    Convert Date  ${current_date_before}  datetime
    \    Log To Console    ${\n}----- Iteration ${iteration_index} [Start] ${time_before.hour}:${time_before.minute}:${time_before.second} -----${\n}
    \    ${cest_result} =    CHECKSET CONNECTION CEST    ${gpib_address}    ${cest_timeout}    ${cest_interval_in_s}
    \    Log To Console    cest_result:${cest_result}
    \    ${result} =    Evaluate    ${zero} - 1
    \    Run Keyword If    "${cest_result}" == "False"
    \    ...    Run Keywords    Log To Console    ${\n}${\n}The IVC is not connecting automatically. Retrying...${\n}    AND    CHECKSET DEACTIVATE CAN    ${can_to_deactivate}    AND    CHECKSET ACTIVATE CAN    AND    CHECKSET IVC STATUS AND    REBOOT    AND    CHECKSET CONNECTION CEST    ${gpib_address}    ${cest_timeout}    ${cest_interval_in_s}
    \    ${result} =    CHECK IVC TO INTERNET    ${internet_address}    ${interface_rf}    ${ct_timeout}    ${inter_ping_time}    ${nb_of_sent_pings}
    \    ${current_date_after} =    Get Current Date  result_format=%Y-%m-%d %H:%M:%S.%f
    \    ${time_after} =    Convert Date  ${current_date_after}  datetime
    \    Log To Console    ${\n}----- Iteration ${iteration_index} [Stop] ${time_after.hour}:${time_after.minute}:${time_after.second} -----${\n}
    \    Log To Console    ${\n}result:${result}${\n}
    \    ${time_after_timeout} =    Add Time To Date    ${time_before}    ${cest_timeout}
    \    ${error_log}    Set Variable If
    \    ...    "${cest_result}" == "False" and "${error_log}" == "ok"    ko: cest timeout of ${cest_timeout} seconds reached > reboot ivc
    \    ...    "${cest_result}" == "True" and "${error_log}" == "ok"    ok
    \    ...    "${error_log}" != "ok"    ko: gpib connection lost > reboot drivers
    \    Log To Console    error_log:${error_log}
    \    Run Keyword If    "${cest_result}" == "True"    SAVE TTC TO FILE    ${time_before}    ${time_after}    ${name_to_save}    ${initial_time}    ${error_log}    ELSE    SAVE TTC TO FILE    ${time_before}    ${time_after_timeout}    ${name_to_save}    ${initial_time}    ${error_log}
    \    ${remaining_iterations} =    Evaluate    ${number_of_iterations}-${iteration_index}
    \    Log To Console    ${\n}${remaining_iterations} iteration(s) remaining...${\n}
    Log To Console    Exited

*** Keywords ***
SETUP_TC_IVC_RELIABILITY_INTERNET
    Log To Console    ${\n}***** Setup Initial Conditions *****${\n}

TEARDOWN_TC_IVC_RELIABILITY_INTERNET
    Run Keyword And Ignore Error    Log To Console    ${\n}***** Teardown *****${\n}
    Run Keyword And Ignore Error    CHECKSET GO TO LOCAL    ${equipment}    ${gpib_address}
```

D.6 TC_CYBERSEC_TLS_MQTT_REJECTS_V10_ACCEPTS_V12.robot

```
#
# 2019 Developed by Renault SW Labs,
# an affiliate of RENAULT s.a.s. which holds all intellectual property rights.
# Use of this software is subject to a specific license granted by Renault s.a.s.
#
*** Settings ***
Documentation     Test to check if the MQTT broker rejects TLSv1.0 but accepts TLSv1.2
...
...               IVC starts TLS SYN
...
Test Setup        SETUP_TC_CYBERSEC_TLS_MQTT_REJECTS_V10_ACCEPTS_V12
Test Teardown     TEARDOWN_TC_CYBERSEC_TLS_MQTT_REJECTS_V10_ACCEPTS_V12
Resource          ../HLK/vehicle_cybersec.robot
Resource          ../HLK/eq_cmw500.robot

*** Variables ***
${file_name}                openssl_client_logs.txt
${timeout_to_mqtt}          6
${interface_rf}             rmnet_data5
${equipment}                CMW500
${gpib_address}             20
${ip_mqtt_avnext}           52.233.180.235
${port_mqtt_avnext}         8883

*** Test Cases ***
TC_CYBERSEC_TLS_MQTT_REJECTS_V10_ACCEPTS_V12
    Log To Console    ${\n}${\n}***** Begin Test Execution *****${\n}
    ${mqtt_connection_status} =    CHECKSET IVC TO MQTT    GMS    ${timeout_to_mqtt}
    Run Keyword If    "${mqtt_connection_status}"== "True"    CHECKSET IVC TO MQTT    DM    ${timeout_to_mqtt}
    CHECKSET FORCE TLS CONNECTION    TLSv1.0    ${ip_mqtt_avnext}    ${port_mqtt_avnext}    ${file_name}
    ${mqtt_connection_status} =    CHECKSET IVC TO MQTT    GMS    ${timeout_to_mqtt}
    Run Keyword If    "${mqtt_connection_status}"== "True"    CHECKSET IVC TO MQTT    DM    ${timeout_to_mqtt}
    CHECKSET FORCE TLS CONNECTION    TLSv1.2    ${ip_mqtt_avnext}    ${port_mqtt_avnext}    ${file_name}
    ${mqtt_connection_status} =    CHECKSET IVC TO MQTT    GMS    ${timeout_to_mqtt}
    Run Keyword If    "${mqtt_connection_status}"== "True"    CHECKSET IVC TO MQTT    DM    ${timeout_to_mqtt}

*** Keywords ***
SETUP_TC_CYBERSEC_TLS_MQTT_REJECTS_V10_ACCEPTS_V12
    Log To Console    ${\n}***** Setup Initial Conditions *****${\n}
    ${status_of_firewall} =    CHECKSET FIREWALL STATUS    ${interface_rf}
    Log To Console    ${status_of_firewall}
    Run Keyword If    "${status_of_firewall}"== "False"    SET IVC FIREWALL    True
    CHECK DEBUGCONSOLE IS CLOSED

TEARDOWN_TC_CYBERSEC_TLS_MQTT_REJECTS_V10_ACCEPTS_V12
    Run Keyword And Ignore Error    Log To Console    ${\n}***** Teardown *****${\n}
    Run Keyword And Ignore Error    DELETE OPENSSL CLIENT LOGS    ${file_name}
    Run Keyword And Ignore Error    CHECKSET GO TO LOCAL    ${equipment}    ${gpib_address}
```

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona